National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

NL-339 (r. 88/04) c

Canada

THE UNIVERSITY OF ALBERTA

AUTOMATIC WINDROWER CUTTER BAR AND REEL HEIGHT CONTROL

by

ROY ALBERT MAKI

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF AGRICULTURAL ENGINEERING

EDMONTON, ALBERTA

SPRING 1989

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-52808-7

Canada

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR ROY ALBERT MAKI

TITLE OF THESIS AUTOMATIC WINDROWER CUTTER BAR AND REEL HEIGHT CONTROL

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1989

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Roy Maki

PERMANENT ADDRESS:

P.O. BOX 361

WAPELLA, SASKATCHEWAN

S0G-4Z0

DATED April 24 1989

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the FACULTY OF GRADUATE STUDIES AND RESEARCH, for acceptance, a thesis entitled AUTOMATIC WINDROWER CUTTER BAR AND REEL HEIGHT CONTROL submitted by ROY ALBERT MAKI in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE.

.....

Supervisor

.....

.....

Date 23 SEP 88

Abstract

A microprocessor based control system was developed for a cereal crop windrower to provide automatic control of cutter bar and reel heights. The control system used a microprocessor-based computer to acquire information from sensors and operator controls, make a control decision, then actuate hydraulic controls and update the operator's console display. Two types of sensors were developed to acquire crop and cutter bar height information. An ultrasonic, non-contact, distance-ranging sensor measured cutter bar-to-ground height and conveyed this data to the microprocessor. A piezo-electric based contact sensor determined crop canopy height position with respect to the reel support frame and coded this information in a binary form that could be read by the microprocessor. Other equipment developed included an operator's console, used to interact with the windrower control process, and an interface junction box, used for a microprocessor-sensor interface and a microprocessor-hydraulic solenoid valve interface.

Tests of the controller were conducted during the 1987 fall harvest season in both wheat and barley crops near D'arcy, Saskatchewan. The cutter bar height sensor was further evaluated for height reading repeatability while travelling over wheat stubble, summerfallow and hay land terrain conditions at the University of Alberta's Ellerslie Research Station in Edmonton, Alberta.

Control performance proved satisfactory under normal crop conditions. Unstable cutter bar control occurred when the crop contained heavy infestations of Russian thistle. Reel control also became unstable when the crop stand was very thin.

Recommendations to improve control and sensor performance are outlined.

Acknowledgements

The author would like to express gratitude to the people who shared their time and resources in this thesis. Recognition is directed to Dr. J. J. Leonard, for his guidance, supervision, and keen interest throughout the project. His unending patience and positive feedback was greatly appreciated.

The author appreciates the National Research Council's funding assistance through the IRAP-H program. Acknowledgement is directed towards Ted Sipos, who acted as the IRAP project manager. His management skills and prompt response in initiating the grant program was appreciated greatly.

The author also expresses recognition to Ross Stonehouse of Innovator Enterprises Inc. for his helpful project input, living accommodations, and the use of equipment and land facilities during field testing.

The technical assistance of Mr. Sandy Yakimchuk and Mr. Ray Holowach, incorporated with the facilities offered through the Department of Agricultural Engineering, contributed greatly to the thesis project. Their enthusiasm and interest made the development and testing stages enjoyable experiences.

To my parents, whose support and encouragement throughout my university career has helped the author more than they will ever know.

Table of Contents

Chapter		Page
1.	INTRODUCTION	1
	1.1 Windrower Design and Operation	2
	1.2 Modern Trends	4
	1.3 Associated Problems	6
	1.4 Requirements for an Automatic Controller	8
2.	PREVIOUS CONTROL DEVICES	9
3.	AIMS	11
	3.1 Development of Crop Height Sensors	11
	3.2 Development of Cutter Bar Height Sensors	11
	3.3 Development of the Controller	11
	3.4 Development of the Operating System Software	12
	3.5 Performance Evaluation	12
4.	WINDROW CONTROLLER DESIGN	13
	4.1 Crop Height Sensor	14
	4.1.1 Other crop sensor technologies	14
	4.1.2 Crop height sensor design	15
	4.1.3 Circuit description	18
	4.2 Cutter Bar Height Sensor	22
	4.2.1 Possible sensor technologies	22
	4.2.2 Ultrasonic sensor theory	23
	4.2.3 Sensor design	24
	4.2.4 Circuit description	30
	4.3 Microprocessor	33
	4.3.1 Microprocessor Hardware	33
	4.3.1.1 Clock and reset	34
	4.3.1.2 Buffers	34

4.3.1.3 ROM and address decoding 34
4.3.1.4 RAM 34
4.3.1.5 ACIA 38
4.3.1.6 PIA 38
4.3.1.7 PTM 42
4.3.1.8 Operator control switch inputs 42
4.3.1.9 LCD 42
4.3.1.10 Sensor input interface 45
4.3.1.11 Hydraulic control output interface 45
4.3.1.12 RS232 communications interface 47
4.3.1.13 Power supply, connections, and cabinetry 47
4.4 Operator's Console 47
4.4.1 Menu operation 50
4.5 Interface Junction Box 51
5. CONTROLLER OPERATING SYSTEM SOFTWARE 52
5.1 Subroutine INIT 52
5.2 Subroutine POLCR 52
5.2.1 Subroutine FIRQ 55
5.2.2 Subroutine IRQ 55
5.3 Digital Display Routines 56
5.4 Data Processing and Control Routines 57
5.5 Data Capture Routine 61
6. TESTING 62
6.1 Materials and Methods 62
6.1.1 Cutter bar control testing methods 63
6.1.2 Reel control testing methods 65
6.1.3 Cutter bar sensor testing methods 66

7. TESTING RESULTS AND DISCUSSION 70
7.1 Reliability 70
7.2 Cutter Bar Height Control 71
7.3 Reel Height Control 77
7.4 Cutter Bar Height Sensor Evaluation 77
8. CONCLUSIONS 82
9. RECOMMENDATIONS FOR FURTHER WORK 83
10. REFERENCES 84
11. APPENDIX A. CONTROLLER OPERATING SYSTEM PROGRAM 87
12. APPENDIX B. CUTTER BAR HEIGHT SENSOR TEST DATA 133

List of Tables

Table		Page
4.1	Sensor output configuration	19
4.2	PIA address locations	38
7.1	Analysis of variance from the cutter bar height sensor evaluation tests	78
12.1	Cutter bar height sensor test results over stubble terrain	134
12.2	Cutter bar height sensor test results over hay land terrain	135

List of Figures

Figure		Page
1.1	Side view of a windrower illustrating control variables	2
1.2	Standard windrower types (adapted from PAMI, 1977c)	3
2.1	Sensor path during a 90° turn using a pull-type windrower	10
4.1	Side view of a windrower illustrating sensor mounting positions	13
4.2	Cross sectional view of the piezo sensor	17
4.3	Crop height sensor electrical signals	19
4.4	Crop height sensor circuitry, part 1	20
4.5	Crop height sensor circuitry, part 2	21
4.6	Cross sectional view of the cutter bar sensor construction	26
4.7	Cutter bar height sensor electrical signals	28
4.8	Polaroid ranging board schematic and parts list (adapted from Polaroid, 1984b)	29
4.9	Cutter bar sensor modification circuitry	31
4.10	Microprocessor, clock, buffer, and reset circuitry	35
4.11	EPROM and address decoding circuitry	36
4.12	RAM and ACIA circuitry	37
4.13	Switch input PIA and debounce circuitry	39
4.14	Sensor sets A and B PIA circuitry	40
4.15	Sensor sets C and D PIA circuitry	41
4.16	PTM gate driving circuitry	43
4.17	PTM interface circuitry	44
4.18	Circuitry to drive hydraulic solenoids	46
4.19	Power supply circuitry for the microprocessor and interface junction box	48
5.1	Main program routine flow chart	53
5.2	Controller memory map	54
5.3	Configuration of the switch status register MODFGS	56

Figure | Page

5.4 Data processing and control routine flow chart 58

7.1 Cutter bar height control using the horizontal sensor and no-pulse control 73

7.2 Cutter bar height control using the vertical sensor and no-pulse control 74

7.3 Cutter bar height control using the horizontal sensor and pulse control 75

7.4 Cutter bar height control using the vertical sensor and pulse control 76

7.5 Cutter bar height sensor response to a step change in elevation 81

List of Plates

Plate		Page
4.1	Unprotected crop height sensor with attached circuit cabinetry	16
4.2	Operator's console	49
6.1	Component placement in tractor cab	64
6.2	Cutter bar and reel sensor orientations and mountings	64
6.3	Horizontal and vertical mounting of the cutter bar sensor	68

1. INTRODUCTION

An important and critical time during the cereal crop production cycle is the harvest operation. Grain quality and yield are dependent on weather, crop condition, and the harvesting method. Cutting and threshing the crop in one operation, or straight combining, is usually employed when crops have ripened evenly and are free of weed problems. Alternatively, windrowing the crop is used as a pre-threshing operation when crop conditions are not conducive to straight combining. Windrowing is the process of clipping off heads along with a portion of the straw stalk, and placing this material in a row on top of the crop stalks or stubble. Windrowing is used to prepare crop material for threshing by allowing the crop to dry uniformly and reduce the risk of losses from shelling, frost, hail, and insects. The windrowing machine is referred to as a windrower or a swather, the term windrower will be used throughout this thesis.

Smith (1964) stated that windrowing is recommended when the following conditions occur:

- grain is unevenly ripened,
- fields are weedy,
- grain is ripe, but straw is green,
- grain is high in moisture content,
- crop conditions are such that legume crops such as Alfalfa and Sweetclover tend to shatter if left until ripe, or
- weather conditions delay direct or straight combining.

In Western Canada, windrowing is an important step in the harvesting of many cereal crops. Windrowing reduces the risk of crop damage and grain losses due to unfavorable weather and the short growing season which are characteristic of the area.

1.1 Windrower Design and Operation

A windrowing machine consists of a knife used to cut the standing crop, a platform canvas or draper used to convey the cut crop material into a windrow, and a reel which is used to direct the crop into the knife and onto the draper. Variables usually controlled by the machine operator include knife or cutter bar height above the ground surface, reel height with respect to the crop canopy top, and reel speed. Figure 1.1 illustrates these control variables. Sickle knife speed and draper travel speed are usually set by the manufacturer and are not adjustable.

A constant speed ratio should exist between reel peripheral speed and ground travel speed (reel speed index) and this should be between 1.1 and 1.2 (PAMI 1977c). Greater values of speed index will damage the crop causing the grain to shell out. Lesser values result in improper placement of the crop material onto the platform draper. Crop material should be placed on the draper parallel with the draper axis of rotation resulting in a properly formed windrow. Four general types of windrows are fantail, herringbone, angled parallel, and parallel (Figure 1.2) which are partially accomplished with proper placement of the crop onto the draper. Most windrowers have variable reel speeds that are adjustable mechanically,

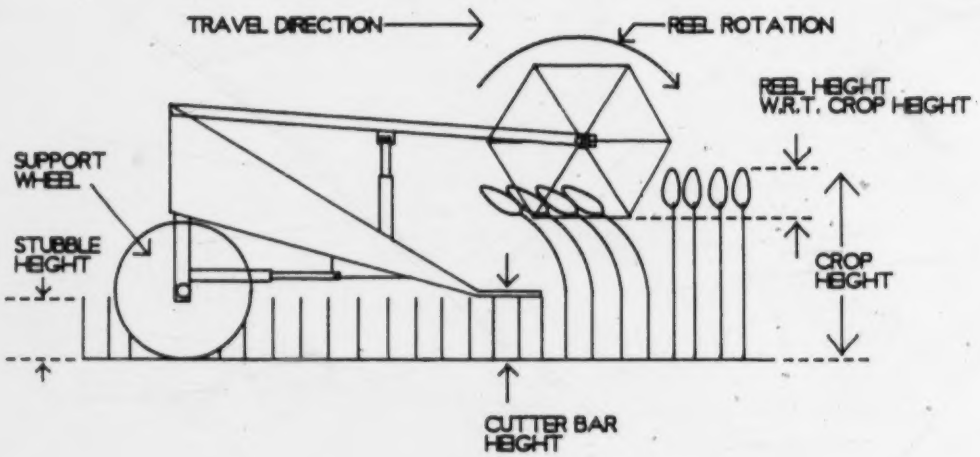


Figure 1.1 Side view of a windrower illustrating control variables

hydraulically, or, in the case of ground driven reels, by varying travel speed.

Reel height is usually set relative to the crop canopy top. The top edges of the reel bats, at the lowest point of travel, should be positioned below the top of the crop heads (Kepner et al. 1978). Variations in reel height caused by improper control may create excessive crop damage and improper placement of the crop material. When the reel penetrates the crop excessively, crop material tends to catch and lift with the reel. This material then is deposited onto the windrower chassis or is propelled by the reel and placed onto the uncut crop canopy. When reel penetration is insufficient, short plants will not be directed onto the platform draper properly. Proper reel height control minimizes crop damage and aids in positioning the crop correctly onto the draper, ultimately producing a suitable windrow.

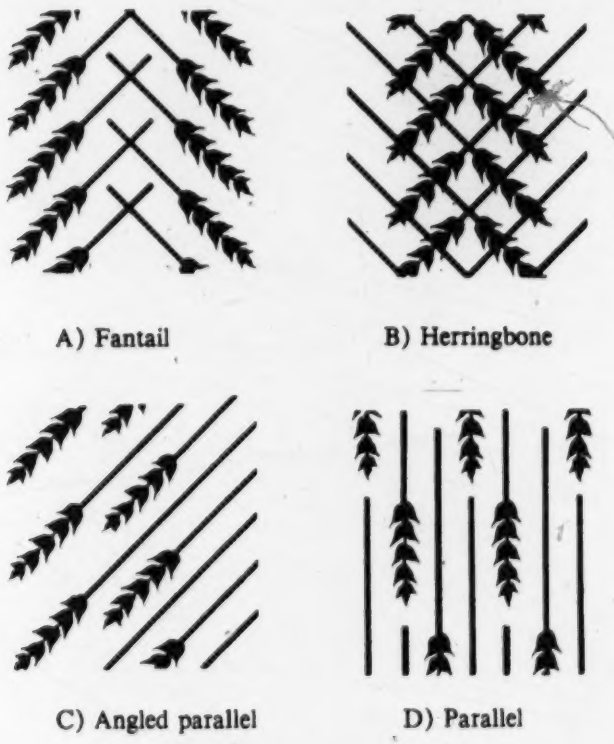


Figure 1.2 Standard windrower types (adapted from PAMI, 1977c)

Optimum cutter bar height above ground or stubble height (the height of the portion of crop stalk left standing after windrowing), is dependent on crop density, crop type, stubble strength, windrower width, and weather conditions (Pask et al. 1973). If stubble is long the windrow is supported high above the ground which enhances drying and ease of pickup. However, long stubble implies a short straw length, which is susceptible to settling through the stubble and onto the ground, lengthening drying time and increasing pickup losses. Short stubble places the windrow too close to the ground surface again making the drying time long and increasing pickup losses. Generally, stubble height is expressed as a ratio of the crop height. Stubble heights one third of the crop height are recommended for average to heavy crops, and one quarter of the crop height for thin crops (Pask et al. 1973). With tall crops, stubble height should not exceed 300 mm. Traditionally, stubble height has been set at the discretion of the machine operator.

Knife speeds are between 600 to 800 cycles per minute, with the stroke being 6 to 10 mm greater than the guard spacing (Kepner et al. 1978). Typical draper speeds range between 2.5 and 3.9 m/s.

1.2 Modern Trends

Combine harvester capacities have increased substantially over the past 10 years. High feed rates of material other than grain (MOG) ranging from 20 to 26 t/h are common capacities of modern harvesters (PAMI 1981a, 1981b, 1982, 1983, 1986, 1987a, 1987b). MOG rates ranging from 12-17 t/h were typical of mid- to late-1970 threshers (PAMI 1977a, 1977b, 1978). Large windrows are required to feed modern harvesters if they are to operate under full capacity at reasonable travel speeds of 3.2-5.4 km/h (ASAE 1983).

Both single-swath and double-swath windrowing techniques are practiced. Single-swath windrowers tend to be large in size with cutting widths as wide as 15 m. Cut crop is placed in single windrows at intervals corresponding to the width of cut. Double-swath windrowing usually employs a smaller windrower, which places crop from two passes in swaths close together enabling a harvester to thresh both swaths at once. An extended

platform draper may be used every second windrow pass to convey the second swath and place it beside the first pass swath (PAMI 1984). Another method commonly employed reverses the draper rotation during the second windrower pass, delivering crop material to the opposite end of the windrower and placing it next to the first swath (Innovator 1987).

Wide, single-swath windrowers have the advantage of being less complex mechanically entailing a less expensive machine to purchase and maintain. However, these windrowers have the inherent problem of maintaining a fixed cutter bar or stubble height when used over variable terrain. To overcome this problem, multi-platform windrowers are used to increase flexibility over hilly conditions. Usually, two or three rigid platform sections with widths of 4.9-7.3 m are pinned together at flexible joints. Each joint then is supported with a transport wheel, used to adjust the cutter bar height hydraulically at the joint (Innovator 1987).

Duplex units with total cutting widths of 15.2 m have been introduced to the windrowing market (Macdon 1986). Two conventional pull-type windrowers are used, the second windrower trailing behind the first windrower. This allows delivery of two windrows, each from a 7.6 m width of cut or one windrow from a 15.2 m width of cut.

With wide multi-platform or duplex windrowing units, stubble shaping is now possible. Stubble shaping is the practice of varying the stubble height across one or two windrower passes. Stubble shaping aids in winter snow catch which enhances the moisture content of the soil for the following crop year (Hoechst 1984).

Stubble shaping also can be employed to minimize the MOG placed in a windrow which increases harvester efficiency. This is accomplished by cutting a shorter straw length at one end of the windrower while cutting a longer straw length at the opposite end. The longer straw length must be placed underneath the windrow in order to support the shorter straw length crop which would settle onto the ground if placed onto the stubble first.

1.3 Associated Problems

Modern trends in windrower design have increased the operational complexity of these machines. Multi-platform and duplex designs require several reel and cutter bar height settings, one for each platform joint, and one for each end of the windrower. The ability of an operator to adjust each height setting properly for varying crop and field terrain conditions becomes impaired as the number of visual judgements and the distance of the operator from the extremes of the machine are increased.

Crop height alone may vary considerably across a field. Terrain, soil, nutrient availability, and moisture variations are all factors that may affect the crop height, and which may vary over short distances within a field. These variations require frequent manipulation of controls to ensure that proper cutter bar and reel heights are maintained across the total width of cut.

When windrowing in level fields, support wheels are adequate for maintaining a proper cutter bar-to-ground distance at a preset level. However, when fields are undulating, transient changes in cutter bar-to-ground height will occur due to the placement of support wheels behind the platform frame. This problem becomes serious when cutting crops close to ground level, as the cutter bar may come into contact with the ground. Damage to the cutter bar may result and lead to costly repairs and the expense of equipment down time.

The ability of a windrower operator to make a correct height adjustment is dependent on the visibility of the windrower cutter bar and reel to the operator, and on the amount of time that the operator has to take appropriate action.

Visibility depends on crop density, dust, lighting, distance of the operator from the point of the machine under observation, and physical obstructions. High density crops inhibit observation of the cutter bar and reel penetration. Dust produced from the crop or field soil can impair visibility of the windrower. Night windrowing, while moisture conditions are suitable, is desirable as the windrowing day can be extended during the critical harvesting stage. Presently, night windrowing depends on artificial lighting mounted on the machine, which is seldom adequate for making proper height judgements. Distance between the

operator and the point under observation can be extensive when operating wide windrowers, making height observations difficult. Obstructions from tractor cab frames or the windrower structure can create blind spots along the cutter bar, which further decreases visibility. Traditionally, when visibility is poor, the operator bases his judgement upon the stubble height behind the windrower after the crop is cut. Hence, heights are adjusted after a deviation occurs, which is not as desirable as being able to adjust heights in anticipation of a change in crop height or terrain.

The time available to an operator to make an adjustment depends on the number of parameters that have to be monitored and the speed of travel. The operator is responsible for the monitoring of windrower parameters, making steering corrections to accurately follow the edge of the standing crop, and watching for hazardous conditions in the crop (e.g. rocks, lodged crops). Large windrowers having multiple control points increase the complexity of the operator's task.

Drayfuss (1960) stated that the total reaction time involved from the time an operator detects an error to the time a control movement is initiated is 1.7 s. Typical windrowing speeds are between 7.9 and 10.8 km/h (ASAE 1983). Thus, during the total reaction time period, the machine can travel 5.1 m. This delay in adjustment due to the complexity of tasks required, reaction time, and operator ability will contribute to inadequate adjustments and will lead, ultimately, to increased grain losses.

Operating a windrower can be tiring after several hours of operation. Constant manual monitoring leads to fatigue and poor judgement, decreasing the quality of the windrowing process. The risk of machine damage due to improper settings is increased by operator fatigue.

Control complexity, operator reaction time, visibility, and operator fatigue all impair the ability of an operator to control the windrowing process properly. This impairment leads, ultimately, to increased grain losses due to improper windrowing. The need for automation of cutter bar and reel height control is, therefore, highly desirable. Not only would automation minimize grain loss, but it would make the operator's task simpler, less tiresome, and capable

of being carried out by semi-skilled personnel.

1.4 Requirements for an Automatic Controller

Several criteria are required when designing a cutter bar and reel height windrower controller.

First, the cutter bar and reel heights should be controlled independently of each other. This allows the operator to select appropriate cutter bar-to-ground or stubble heights without affecting the reel position. Also, the reel height with respect to crop height can be adjusted without affecting the stubble height setting.

Second, the cutter bar and reel height setpoints should be changeable while windrowing. Interaction with the control process should not interrupt the windrowing operation, but allow changes to be made "on the go", which eliminates down time due to adjustments.

Third, both automatic and manual control modes should be operator selectable. Manual control initiated by the operator is required when hazardous conditions occur in the crop (e.g. rocks, lodged crops), or when transporting the machine from field to field.

Fourth, the operators console should promote "user friendliness". The control panel should be easy to use and read while operating the windrower.

Fifth, the controller should be suitable for large multi-platform windrowers, capable of controlling four cutter bar and four reel height positions. A controller of this size should meet the requirements for most large multi-platform windrower designs.

Sixth, the controller hardware should be able to be interfaced easily with existing windrower equipment. Adaptation of automatic control to a windrower should require only minor modifications to the existing machine.

Seventh, the controller should be powered from a 12V power system, which is standard on most windrowing equipment.

This thesis describes the design, testing, and field evaluation of an automatic cutter bar and reel height controller that was developed to satisfy the above requirements.

2. PREVIOUS CONTROL DEVICES

Automatic control of windrowers has been studied during the past 23 years. Kaminski and Zoerb (1965) recognized the need for cutter bar height control and constructed a prototype controller. The sensing unit was mounted on the cutter bar and consisted of eight lever-actuated microswitches to detect crop and cutter bar height. A shoe, which slid along the ground, was used to aid in cutter bar height adjustment. The eight microswitches controlled a hydraulic solenoid valve used to raise and lower the cutter bar platform. This controller accomplished three-position, on-off control of cutter bar height only. The device performed satisfactorily, but sensor components were prone to excessive wear. Reel height was not controlled in this design.

Pask et al. (1973) developed a mechanical-electrical control device that adjusted the cutter bar height in proportion to the crop height. The sensor consisted of two lever-actuated microswitches mounted above the platform draper and behind the reel. The sensor then was positioned by a cable and pulley system in relation to the cutter bar height. Sensing was accomplished by detecting the crop after being cut, as it fell onto the sensor. Three-position, on-off control of the cutter bar height was accomplished using a solenoid-hydraulic valve along with analog electrical circuitry for filtering and control. Satisfactory results were obtained from this system, but the mechanical sensors were prone to wear, and the complex mechanical cable and pulley system would be difficult to install on most types of windrowers. Also, a ground proximity sensor, mounted beneath the cutter bar, and used to prevent cutter bar-ground contact, was susceptible to damage while under reverse and turning motions. This is an important consideration in the design and location of sensors on windrowers. Figure 2.1 depicts the path and direction that a cutter bar height sensor would travel during a 90° turn when using a pull-type windrower. Crop height was not a control variable in this design.

Senstek (1987) has introduced a commercially-available cutter bar height control for harvesters and windrowers. Cutter bar height is measured using a microprocessor controlled ultrasonic sensor. Cutter bar height can be controlled automatically within a height range of 76 to 560 mm. A performance evaluation of this controller was not available. Reel height was

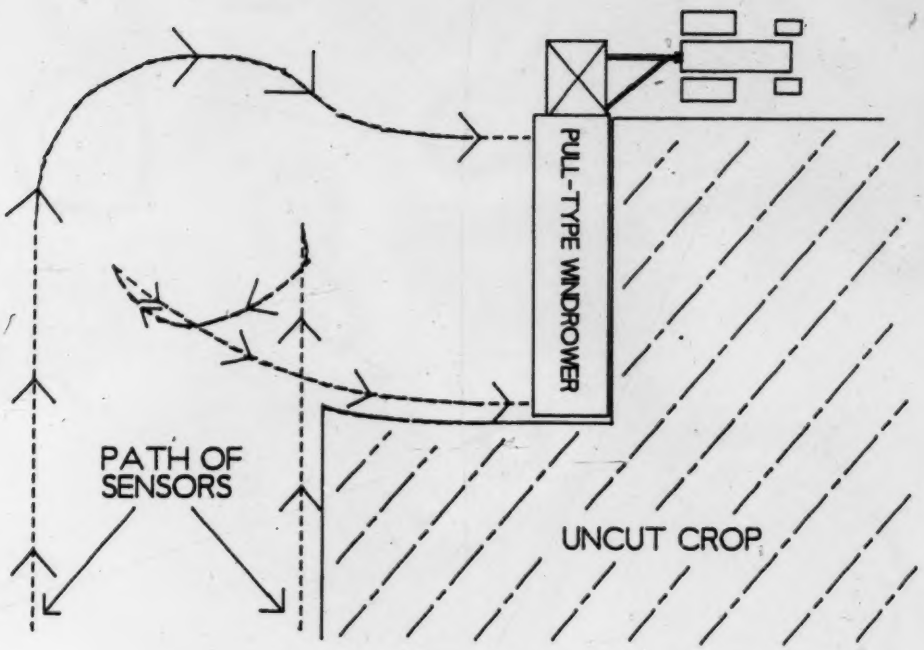


Figure 2.1 Sensor path during a 90° turn using a pull-type windrower

not controlled with this device.

Thornley et al. (1985) have been granted a patent on an automatic height control system for a header on a farm implement. Crop height is determined by ultrasonic detectors, and, by means of electronic and electro-mechanical systems, the table is adjusted to a proper height with respect to the crop height and ground level. Gauge wheels together with a mechanical linkage are used to sense and control the cutter bar height above the ground. Thus, cutter bar height is controlled on a self-proportioning basis with respect to crop height. With this design, the crop canopy top may not be detected properly with thin crop stands. Also, the mechanical gauge wheels may sustain damage when the machine is turning. Again, the complexity of the mechanical equipment needed may inhibit its adaptability to some windrower or header equipment. Details of the performance of this device have not been published.

3. AIMS

The overall aim of this project was to develop an automatic cutter bar and reel height control system that would satisfy the requirements outlined in section 1.4.

More specifically, five primary objectives were identified.

3.1 Development of Crop Height Sensors

The crop height sensor should be capable of determining the position of the crop canopy top with respect to the reel support frame and relay this information to the controller. The sensor should also distinguish between crop and no crop conditions. No crop conditions occur when finishing a field, making a turn, or the windrower not operating. The sensor should withstand normal operating conditions, being unaffected by vibration, moisture, and dust.

3.2 Development of Cutter Bar Height Sensors

The cutter bar height sensor should be capable of measuring the cutter bar-to-ground distance and relaying this information to the controller. Also, the sensor should provide the controller with temperature compensated data. The sensor should operate under normal conditions, being immune to moisture, dust, and vibration.

3.3 Development of the Controller

The controller should provide a controller-sensor, controller-machine, and controller-operator interface, which are essential elements of data acquisition and control. An operator's console was required for the controller-operator interface, and an interface junction box required for the controller-sensor and controller-machine interface. The controller should be microprocessor based so that a software operating system could be utilized for windrower control. The controller hardware should operate under normal windrower operating conditions providing immunity to vibration and electrical interference.

3.4 Development of the Operating System Software

The operating system software should schedule a continuous loop of tasks required for windrower control. The operating system must read four reel height and four cutter bar height sensors, read operator control switches, update control of the machine hydraulics as required, and display control information to the operator via the operator's console. Reel height should be controlled such that the reel follows the top of the crop canopy. The cutter bar should be controlled within ± 50 mm of a user programmed setpoint. Response time of the controller to compensate for a change in terrain height or crop height should range from 0.5 to 1 s. Error checking of sensor information should also be carried out by the operating system software.

3.5 Performance Evaluation

Performance evaluation of the controller was a further objective of the project. Cutter bar and reel height control were to be evaluated in the field on the basis of control stability and ability to follow a setpoint.

The ability of the cutter bar height sensor to measure the sensor-to-target distance was evaluated under a range of operating conditions which would be encountered in the field.

Crop height sensor control should be judged on its ability to determine the top of the crop position and thus control the reel height such that the top edges of the reel bats, at the lowest point of travel, should be positioned below the top of the crop heads.

Design, testing, and evaluation of the above objectives are presented in the following chapters.

4. WINDROW CONTROLLER DESIGN

The controller was designed to accommodate large windrower machines capable of controlling four sets of hydraulics independently. Each set of hydraulics was controlled with two sensors: a cutter bar height distance sensor and a crop height sensor. Figure 4.1 shows the location of sensors on a windrower machine. The cutter bar height distance sensor was positioned in front of the cutter bar, and measured the distance between the ground and the height of cut. This measurement was used to update cutter bar position by raising or lowering the equipment's platform hydraulics. The crop height sensor was fixed onto the reel support frame and positioned in front of the reel, ensuring sensor contact with the uncut crop. The crop height sensor determined the position of the top of the crop relative to the reel frame. This information was used to position the reel at the proper height with respect to the crop top, utilizing the windrower's hydraulic reel positioning system. An operator's console allowed the operator to control and monitor the windrowing process. The operator could adjust and view all cutter bar and reel height setpoints while operating the windrower. Automatic and manual control switches for both reel and cutter bar control were supplied. A microprocessor was utilized to acquire data from the sensors, make a control decision based upon the setpoint

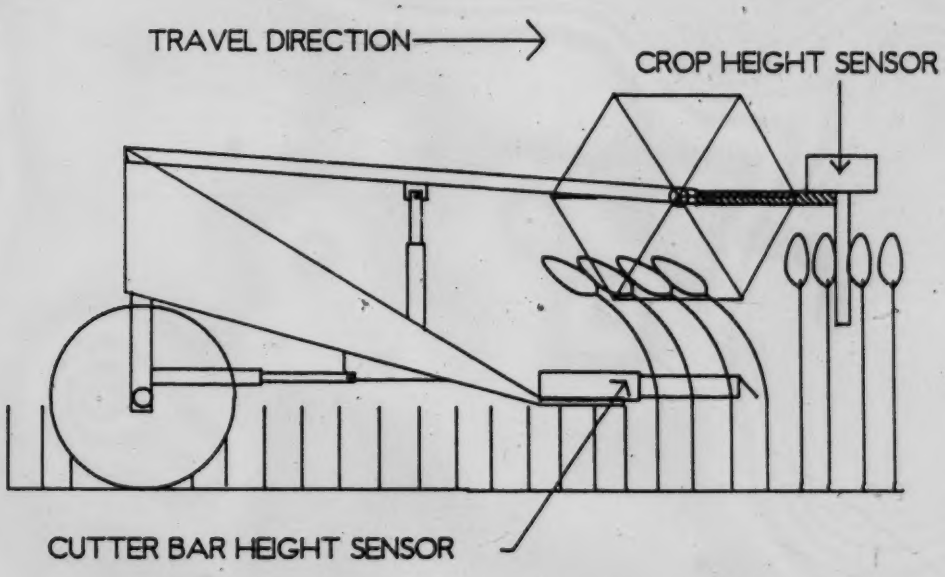


Figure 4.1 Side view of a windrower illustrating sensor mounting positions

and actuate the appropriate hydraulic circuits as required. The device consisted of four crop height sensors, four cutter bar height sensors, a microprocessor controller, operating system, an operator's console and an interface junction box. These components are described in detail below.

4.1 Crop Height Sensor

The crop height sensor was a contact sensor employing a piezo-electric foil (Kynar, Pennwalt Corporation, King of Prussia, Pennsylvania) used to detect crop impact. The sensor was fixed onto the reel frame and extended in front of the reel providing sensor contact with the standing crop.

The piezo-electric foil consisted of a polymer piezo film, 26 μm thick, in between two conductive sheets of aluminum foil. When physical deformation caused a dynamic thickness change in the piezo film, an electrical potential appeared across the two conducting materials. Thus, a mechanical-electrical transducer was formed. Impact of crop upon the piezo-electric foil surface was sufficient to produce deformation causing a voltage impulse output. The amplitude and duration of the impulse was dependent on the amount and speed of deformation.

4.1.1 Other crop sensor technologies

The piezo-electric contact sensor was utilized because of simplicity and cost effectiveness. Other detection principles considered were mechanical, infrared-light reflectance, beam interruption, capacitance, and ultrasonics.

Mechanical sensors have the inherent problems of physical wear and crop interference. Mechanical sensors also tend to be directional, producing inaccurate readings when the windrower moves backwards or sideways during a turn. Also, such movements may damage the sensor.

Detection of crop using light could be accomplished either by light reflection or beam interruption. Light reflection works on the assumption that light from infrared sources will be

reflected off the crop surface, with the amount reflected quantified with a light detector. However, different crops would have different reflective characteristics and calibration would be required for each crop variety. Also, commercially available units are not economically feasible. Beam interruption utilizes emitter-detector pairs, which produce an electrical signal when material passes between the sensor pairs. Implementation of this concept would require several emitter-detector pairs positioned in a vertical array and placed in front of the reel within the standing crop. The physical size and position of the emitter-detector pairs would prove undesirable because of crop interference. Also, light sensors are affected by dust and background light sources, which are characteristic of the windrowing environment.

Capacitance proximity sensors determine distance to a target by measuring the flow of an alternating current between two capacitor plates, the sensor acting as one conductor, and the target as the other (O'Sullivan,1985). The characteristics of target-to-transducer capacitance are dependent upon target-to-transducer distance, as well as the type and condition of the crop. The characteristics of most crops are not consistent as row spacing and crop density are never constant. Also, the dielectric constant changes with crop chemical applications, crop moisture content, and variable weather conditions.

An ultrasonic proximity transducer operates on the principle of sound reflection. Knowing the speed of sound, the transducer-target distance can be quantified by measuring the elapsed time between a signal transmission and reception of the reflected signal. The success of this transducer to measure a distance is dependent on the target's ability to reflect a sound wave. Under heavy crop conditions, reflectance of the top of the crop canopy would be adequate. However, when the crop is thin, sound reflectance quality deteriorates and the crop canopy top may not be sensed.

4.1.2 Crop height sensor design

The contact sensor used consisted of four, 20 mm wide strips of piezo film wrapped around 25 mm square mild steel tubing. The four strips were placed 80 mm apart, measured from strip center to center. Figure 4.2 illustrates a cross sectional view of the tubing with the

piezo strips. Plate 4.1 shows the completed sensor.

A minimum of two strips were required to sense the crop canopy top as reel height was controlled so that the top of the crop remained in between two piezo strips. Four strips were used so that the operator could select 1 of 3 reel height setpoints in 80 mm increments while the machine was in operation. To protect the piezo film from damage due to environmental effects such as moisture, sunlight, and wear produced from crop contact, a rubber sleeve was placed over the sensor probe.

Electronic circuitry was designed to determine which of the four piezo strips had impacted with the standing crop (Figures 4.4 and 4.5). Impacts on the piezo strips caused by the standing crop are discrete random events. The design logic of the sensor circuitry provided the microprocessor with continuous information on the number of piezo strips in contact with the standing crop.

The task of providing a continuous output based on discrete random inputs required some basic assumptions on crop-sensor impacts. To eliminate unrepresentative inputs caused

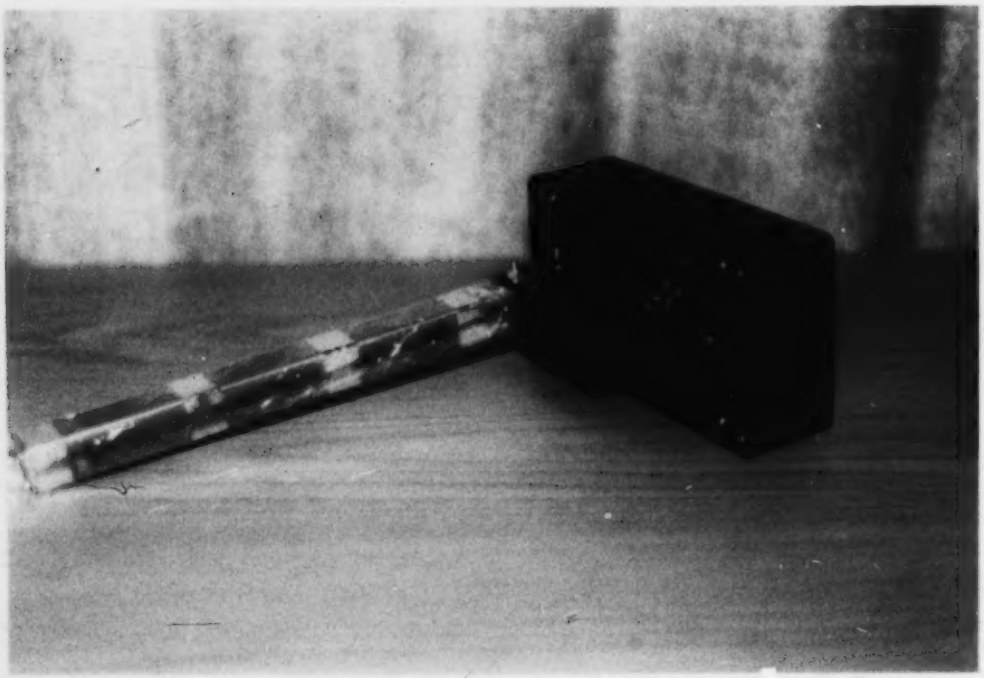


Plate 4.1 Unprotected crop height sensor with attached circuit cabinetry

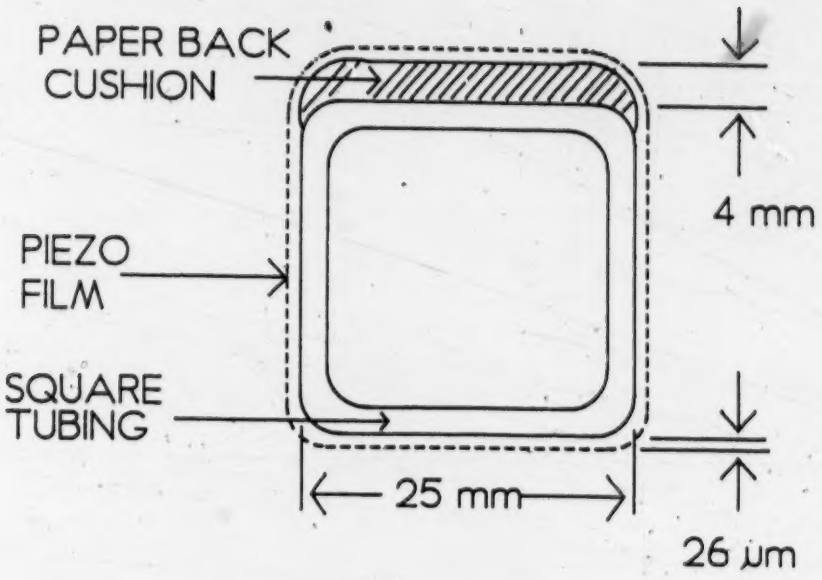


Figure 4.2 Cross sectional view of the piezo sensor

by impact of crop material taller than the average crop height, four impacts on a piezo strip within a predetermined time limit were assumed to be required for a valid input to be recognized by the microprocessor. The time limit could be varied from 0 to 1 s by adjusting a potentiometer provided on the sensor circuitry.

Once a piezo strip received four crop impacts within the appropriate time period, that piezo strip was recognized as a valid input and a digital output signal was made available for the microprocessor to poll. However, once a piezo strip was recognized, a time-out sequence, which measured the time between crop impacts, was required to de-latch the piezo strip to an "off" position if crop impacts ceased. This time-out sequence could be varied from 0 to 1 s by adjusting the same potentiometer as stated above. The sensitivity of each piezo strip could be adjusted to suit different crop impact strengths. This was accomplished by adjusting a potentiometer provided on the sensor circuitry which changed the gain of the piezo sensor amplifiers.

When the windrower was making a turn, finishing a field, or stopped, the crop height sensor would not register crop impact signals. Thus, the reel would lower down to the lowest position. To rectify this problem a microswitch was placed directly behind the cutter bar knife. When crop was being cut, the crop material contacted the microswitch causing it to close contacts. When crop was not being cut, the microswitch would remain open and the sensor would provide the microprocessor with an appropriate input to inhibit automatic reel height adjustment.

Sensor information was made available to the microprocessor in digital form. A sensor-microprocessor interface consisted of three binary information lines which provided eight different binary combinations, six of which were used by the sensor. Table 4.1 lists the sensor output states used.

4.1.3 Circuit description

Figures 4.4 and 4.5 are diagrams of the crop height sensor circuit. An LM324 quad operational amplifier with variable gain was used to amplify the piezo strip inputs. Output from the LM324 was used to trigger a 555 timer set to operate in single-shot, monostable mode. The output of the 555 timer produced a single high pulse of 100 ms duration on every impact and was used to trigger a 74LS90 decade counter. The monostable pulse was required to filter multiple input signals which occur with a single piezo strip-crop impact. Figure 4.3 illustrates the signal descriptions.

Likewise, a second 555 timer was triggered by the operational amplifier output. This timer was used to produce the time-out sequence that could be varied from 0 to 1 s by utilizing the potentiometer. Timer output remained high until the time period between piezo sensor impacts exceeded the predetermined time-out sequence.

If the 74LS90 decade counter had accumulated a count of four and the output from the second timeout 555 timer was in a high state, then the piezo sensor was recognized as a valid input and a latch signal was sent to a 74LS148 priority encoder. If this condition was not met, the decade counter was reset and the process repeated.

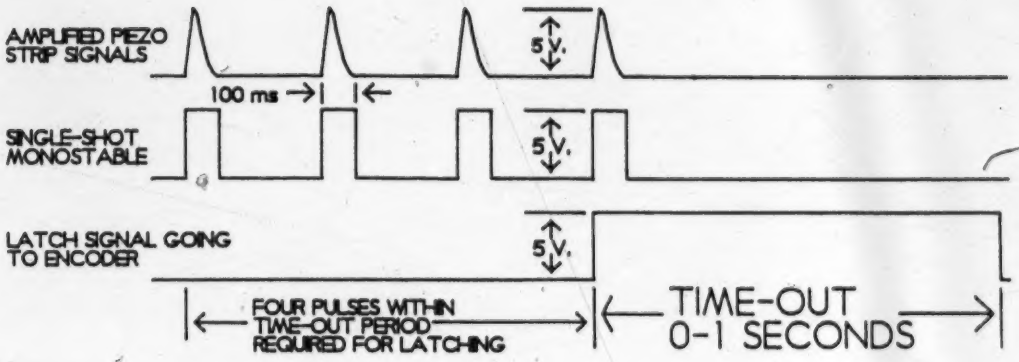


Figure 4.3 Crop height sensor electrical signals

Table 4.1 Sensor output configuration

Data line states			Sensor state
D0	D1	D2	
0	0	0	Zero crop impacts
1	0	0	Piezo strip #1 impacted
0	1	0	Piezo strip #2 impacted
1	1	0	Piezo strip #3 impacted
0	0	1	Piezo strip #4 impacted
1	1	1	Windrower not operative

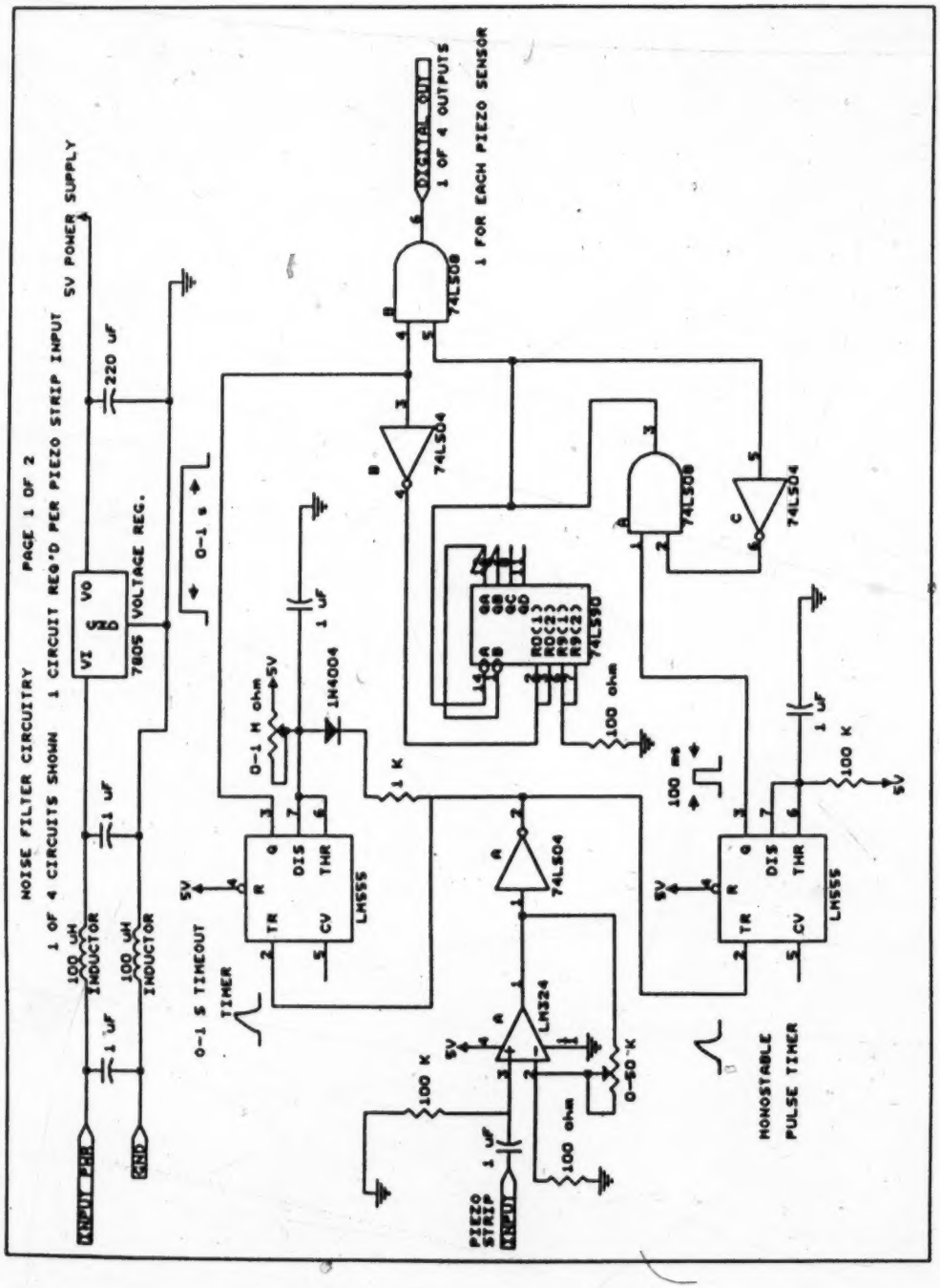


Figure 4.4 Crop height sensor circuitry, part 1

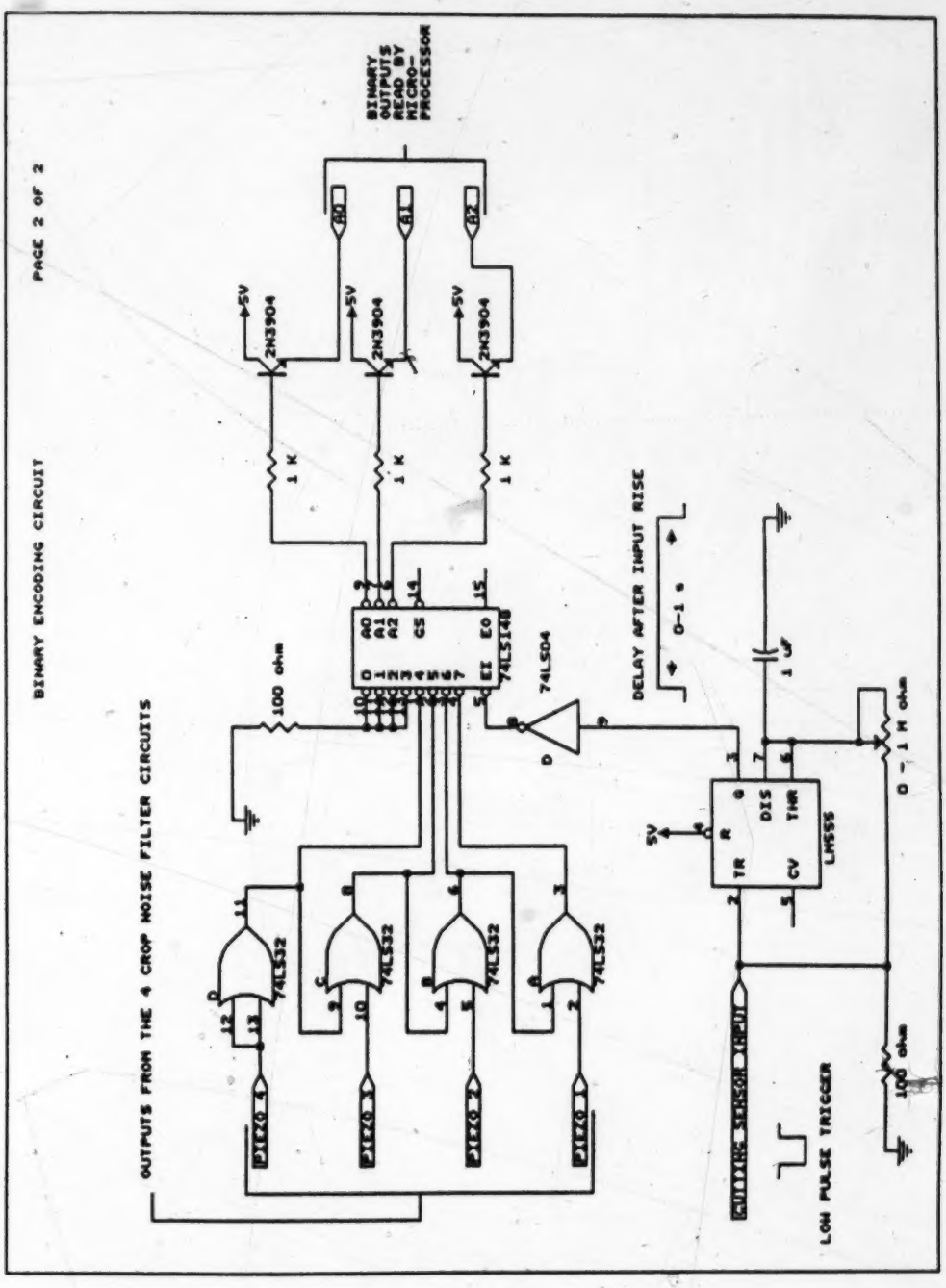


Figure 4.5 Crop height sensor circuitry, part 2

The priority encoder produced a three-bit binary output, the states being determined by eight input signals, five of which were used. Four inputs were latch signals corresponding to the four piezo sensors. The fifth input was supplied by the output of a 555 timer which was triggered by the crop activated microswitch. As the microswitch would be triggered by discrete random events, a variable time-out length also was required so that the timer output remained latched until the time between microswitch closures, caused by crop contact, exceeded a predetermined time. Time-out could be adjusted between 0 and 1 s with a potentiometer provided in the circuitry. The three binary outputs from the priority encoder were current-amplified and read by the microprocessor according to the logic scheme depicted in Table 4.1.

Other circuity consisted of 5V power regulation and a π section R-C filter used to attenuate voltage spikes. Power for the sensor was supplied by the interface junction box. A printed circuit board was made and populated with the required components. Protective cabinetry was provided to give protection from environmental hazards such as moisture and dust.

4.2 Cutter Bar Height Sensor

The cutter bar height sensor used ultrasound to measure the distance from the cutter bar to the ground. The sensor was fixed to the windrower platform and placed in front of the cutter bar.

4.2.1 Possible sensor technologies

Other sensing methods reviewed were infrared light reflectance and mechanical contact. Kirk and Krause (1975) implemented an infrared photoelectric proximity sensor to detect crop edge in order to steer a windrower automatically. Such a sensor is dependent upon the color of the crop as more infrared light is reflected from a light colored target than from a dark colored target. Although the sensor is non-contact in nature, use of such a sensor would require continuous calibration as soil color is variable.

Mechanical contact sensors create crop interference problems if placed in front of the cutter bar. When the windrower backs up or moves sideways on a turn, mechanical sensors may sustain damage. Mechanical sensors also tend to be difficult to install on existing equipment.

Coad et al. (1979) investigated the use of ultrasonics for height control on sugarcane harvesters. Results indicated that cutting height regulation using ultrasonic detection was feasible and accurate within ± 3 mm of the true height. Gunderson et al. (1981) compared ultrasonic, ski, and ski-wheel systems for tillage depth measurement. The ultrasonic system compared favorably to the other mechanical systems but had the distinct advantage of being non-contact. Thus an ultrasonic sensor was used for cutter bar-to-ground distance measurement because of its cost-effectiveness and non-contact sensing principle. Ultrasonics provides the accuracy required for cutter bar height control.

4.2.2 Ultrasonic sensor theory

Distance between an ultrasonic sensor and a target is indicated by the time elapsed between the transmission of an ultrasonic pulse and the reception of an echo. Knowing the speed of sound, the distance can be calculated with the following formula.

$$D = V\ t\ 1000/2 \qquad 4.1$$

where: D = sensor-to-target distance (mm)

V = speed of sound (m/s)

t = elapsed time between transmission and reception (s)

Accuracy of this calculation is dependent upon knowing the speed of sound, which is a function of air density, which, in turn, is a function of moisture content, temperature, and barometric pressure. Of these, temperature has the greatest effect. Deviations in temperature affect the speed of sound according to the following formula (Singh et al. 1984)

$$V_1/V_2 = \sqrt{(T_1/T_2)} \qquad 4.2$$

where: V_1 = speed of sound at T_1 (temperature 1) (m/s)

V_2 = speed of sound at T_2 (temperature 2) (m/s)

T_1 = temperature 1 (°K)

T_2 = temperature 2 (°K)

This calculation illustrates a 5.0 % inaccuracy for a 30 °C variation in temperature, which can occur during the windrowing season. Changes in air densities caused by moisture content and barometric pressure have relatively minor effects on the speed of sound causing variation of only 0.35 % at 20 °C (Polaroid 1984a).

To eliminate the error produced by variations in the speed of sound, two targets were used producing two return echos. The first target was positioned at a known distance from the ultrasonic transceiver, with the second target being the object whose distance to the sensor was to be determined. Knowing the distance, D_1, from the first target to the ultrasonic receiver, and measuring the two echo times t_1 and t_2 corresponding to the first and second target respectively, an environmentally-compensated distance to the object in question can be calculated with the following formula.

$$D_2 = D_1 t_2/t_1 \qquad 4.3$$

where: D_2 = the sensor-to-target distance (mm)

D_1 = transceiver-to-first target distance (mm)

t_1 = first echo return time (s)

t_2 = second echo return time (s)

This calculation remains accurate as long as air conditions are similar between the two targets.

4.2.3 Sensor design

The sensor design was based on a commercially available ultrasonic sensor (Model 606191 Polaroid Corporation). This unit was chosen because of its cost-effectiveness and availability.

Two components comprise the Polaroid Ultrasonic Ranging Unit: an acoustical transceiver transducer and a ranging circuit board. The acoustical transducer consists of a Kapton plastic foil having a gold conductive coating on the exposed side, while the other side

of the foil is stretched over an aluminum back plate. The backplate forms a capacitor that, when charged, exerts an electrostatic force on the foil causing movement. The acoustical transceiver transforms electrical energy into sound energy during the transmit phase, and transforms sound energy into electrical energy during the receive phase. The Ultrasonic Ranging circuit board interfaces with the acoustic transducer. When activated, the unit produces a burst of 56 pulses at an ultrasonic frequency of 49.1 KHz (Polaroid 1984b). The duration of this pulse train is 1 ms. The burst of 300 Vpp pulses is sent to the acoustic transducer, which transforms the energy into sound. Once the pulse train has completed its cycle, the circuit monitors the transducer for echo signals. The energy contained in the returning sound wave is low in comparison to the energy transmitted. Thus, a considerable amount of amplification is required to detect echos. Furthermore, sound energy decreases with distance travelled because of the dispersion of the sound wave in an air media. To compensate for this decrease in reflected sound energy, the ranging system increases amplification with elapsed time.

To obtain environmental compensation, the ultrasonic transducer was placed in one end of a length of 38 mm I.D., schedule 80 PVC pipe (Figure 4.6). When a sound wave is transmitted inside an enclosure such as a pipe, reflections of the sound wave occur at both ends of the enclosure and produce a non-propagating standing wave. Sound is reflected because of a large change in characteristic impedance which occurs at the outlet of the enclosure. To alleviate sound reflection, a wedge was cut out of one end of the pipe at a 45° angle to the pipe longitudinal axis. This shape allowed the ultrasonic transducer to be mounted easily into the pipe, and, at the same time, damp out internal sound reflection by creating a gradual change in characteristic impedance. The other end of the pipe was cut square and produced a sound reflection. This reflection emulated the first target which was used for environmental compensation. The effective distance D_1 to the first target (or distance from the transducer to the end of the pipe) was 295 mm. This distance, (D_1) corresponds to an echo transmit-receive elapse time of approximately 1 ms. The polaroid ranging circuit board requires a 1 ms time delay (after transmission of the sound pulse) before it is capable of

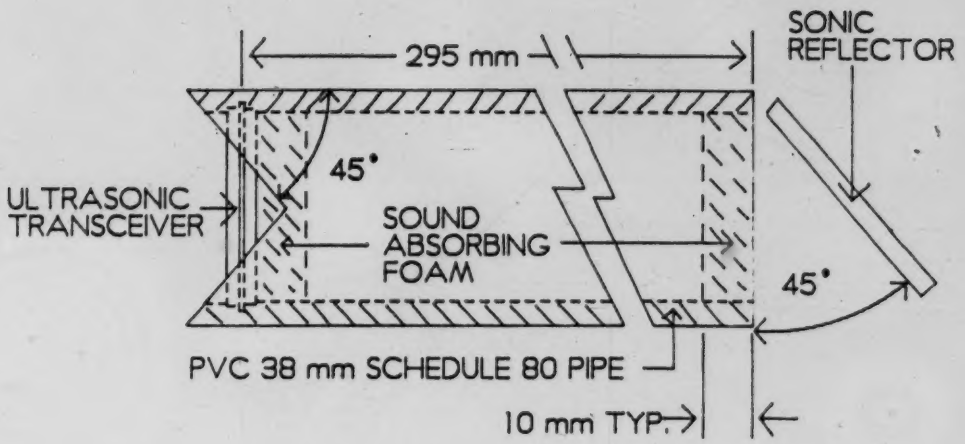


Figure 4.6 Cross sectional view of the cutter bar sensor construction

detecting an echo. This distance ensures that echos will be received after the time delay has elapsed.

As shown in Figure 4.1, the sensor was designed to be mounted in front of the cutter bar. Due to the length of the pipe, a sonic reflector placed at a 45° angle with the pipe was used to deflect the sound pulse towards the ground, allowing the transducer to be mounted horizontally. The sonic reflector consisted of a smooth flat piece of plexiglass mounted in front of the end of the pipe. Unfortunately, unwanted echos were produced by the sonic reflector due to irregularities of the reflector construction. To attenuate these unwanted echos, a thin layer of sound absorbing foam was placed directly in front of the ultrasonic transceiver and at the exit end of the pipe (Figure 4.6). Thus, the foam cover acted as an acoustic filter, which only allowed primary sound echos to be returned while damping weak secondary echos. The foam cover has the additional advantage of preventing dust and moisture from entering the pipe. Also, the ultrasonic sensor could be operated vertically by removing the sonic reflector. Vertical operation of the sensor could be initiated if mounting did not create interference problems with the reel or the crop.

Figure 4.7 shows the waveforms describing the operation of the ranging circuit board. The VSW signal initiated the transmit/receive cycle by stepping from 0 to 5V. After a 5 ms time delay, the 56 pulses were transmitted as displayed by the XLG line. The transmit pulse, as well as the unprocessed amplified echos, appeared on pin 6 of the analog IC (U1). Pin 9 of the analog IC (U1) provided the processed echo wave forms. Figure 4.8 shows the ultrasonic ranging circuit diagram.

Modifications to the ranging circuit board were required to allow environmental compensation and provide an interface to the microprocessor data acquisition system. The ranging board supplied a 1 ms transmit pulse which produced a 1 ms duration echo. The length of this echo corresponded to a sound wave propagation distance of about 280 mm. When two echos were required for environmental compensation, the second echo, corresponding to the object whose distance is in question, would not be detected unless the target was farther than 280 mm from the end of the pipe, where the first echo occurred. To rectify this problem, the transmit pulse was shortened to 0.1 ms or 4 pulses. This allowed secondary echos to be detected from objects being as close as 30 mm from the end of the pipe, which was acceptable for proper operation of the cutter bar sensor.

The second modification to the ranging circuit board consisted of pulse formation circuitry which transformed analog echo signals produced by the ranging circuit board into digital pulse duration data, which the microprocessor could interpret. Figure 4.7 shows the waveforms produced by the modified circuitry. The two digital signals which were read by the microprocessor represent the elapsed time between the transmission of the sound pulse and the returned echos. This time period was represented by the duration that the digital signal remained in a high state.

The first echo digital signal underwent a low-to-high transition when the transmission of the sound pulse began and remained high until a received echo was detected which produced a high-to-low transition. The first echo pulse duration represented the time required for a transmitted sound wave to travel to the end of the pipe and back to the transducer.

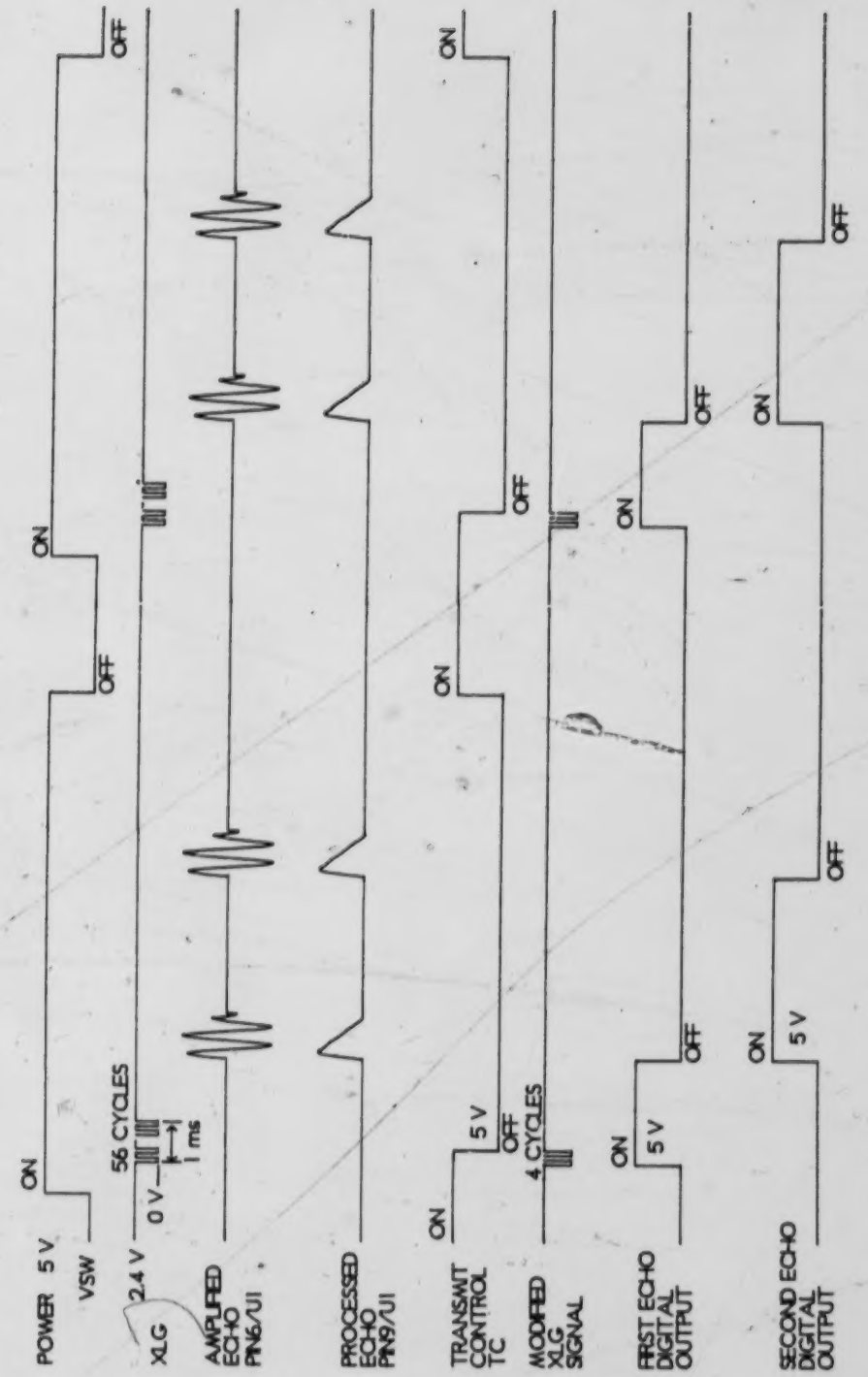
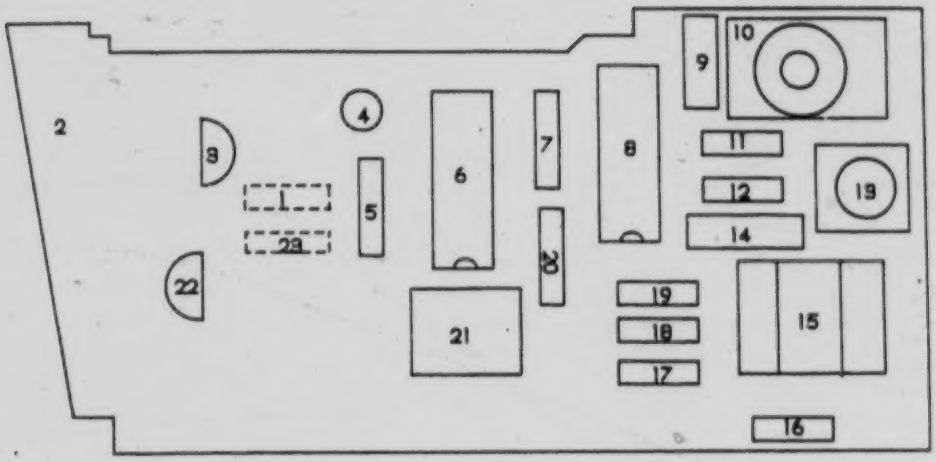


Figure 4.7 Cutter bar height sensor electrical signals



ITEM	REF.	DESCRIPTION
1	R10	22K RESISTOR
2		P.C. BOARD
3	Q1A	2N-4401 TRANSISTOR
4	C3	1 uF CAPACITOR
5	R9	150 ohm RESISTOR
6	U2	DIGITAL IC
7	C4	.001 uF CAPACITOR
8	U1	ANALOG IC
9	C1	.01 uF CAPACITOR
10	L1	TUNED CIRCUIT INDUCTOR
11	C2	.01 uF CAPACITOR
12	R4	62K RESISTOR
13	R6	25K VAR RESISTOR
14	C5	.0022 uF CAPACITOR
15	T1	TRANSFORMER
16	R1	10K RESISTOR
17	CR1	1N4006 DIODE
18	CR2	1N4006 DIODE
19	R2	130K RESISTOR
20	C10	.01 uF CAPACITOR
21	XTAL	420 kHz CRYSTAL
22	Q2A	MPS-A14 TRANSISTOR
23	R11	2.2K RESISTOR

Figure 4.8 Polaroid ranging board schematic and parts list (adapted from Polaroid, 1984b)

The second echo digital signal underwent a low-to-high transition when the first echo made a high-to-low transition (first echo received signal) and remained in a high state until a second echo was detected which produced a high-to-low state change. The second echo pulse duration measured the time elapsed between first-echo-detect and second-echo-detect analog signals.

The two echo pulse durations were measured by the microprocessor. Thus, the target distance from the end of the sensor pipe was calculated by the microprocessor using the following equation.

$$D_2 = D_1((t_1+t_2)/t_1 - 1.0) \qquad 4.4$$

where: D_2 = the distance from the end of the pipe to the target in question (mm)

t_1 = first echo pulse duration time (ms)

t_2 = second echo pulse duration time (ms)

D_1 = distance from the transducer to the end of the pipe (mm)

4.2.4 Circuit description

Circuitry used to reduce the 56 300 Vpp ultrasonic cycles down to 4 cycles utilized a decade counter, a flip-flop and logic gates. The circuit diagram is depicted in Figure 4.9. The transmission line (XLG), used by the ranging circuit board, was used as a clock input on a 74LS90 decade counter configured for binary coded decimal (BCD) count. After 4 XLG transmission cycles, the Q1 output from the decade counter triggered the clock input of a 74LS76 J-K flip-flop configured in toggle mode, which then produced a high-to-low state change on the inverted Q flip-flop output. The inverted Q signal, designated by TC (transmit control) in Figure 4.7, was used to pull the collector voltage of transistor Q1A (Figure 4.8) low terminating the transmit sequence. The inverted Q signal remained low until the J-K flip-flop and the decade counter were reset by a negative-edge-input/negative-pulse-output circuit, which, in turn, was triggered by the VSW signal supplied by the microprocessor. The negative-pulse-output was created using a 74LS08 AND gate in conjunction with an RC circuit used as a trigger.

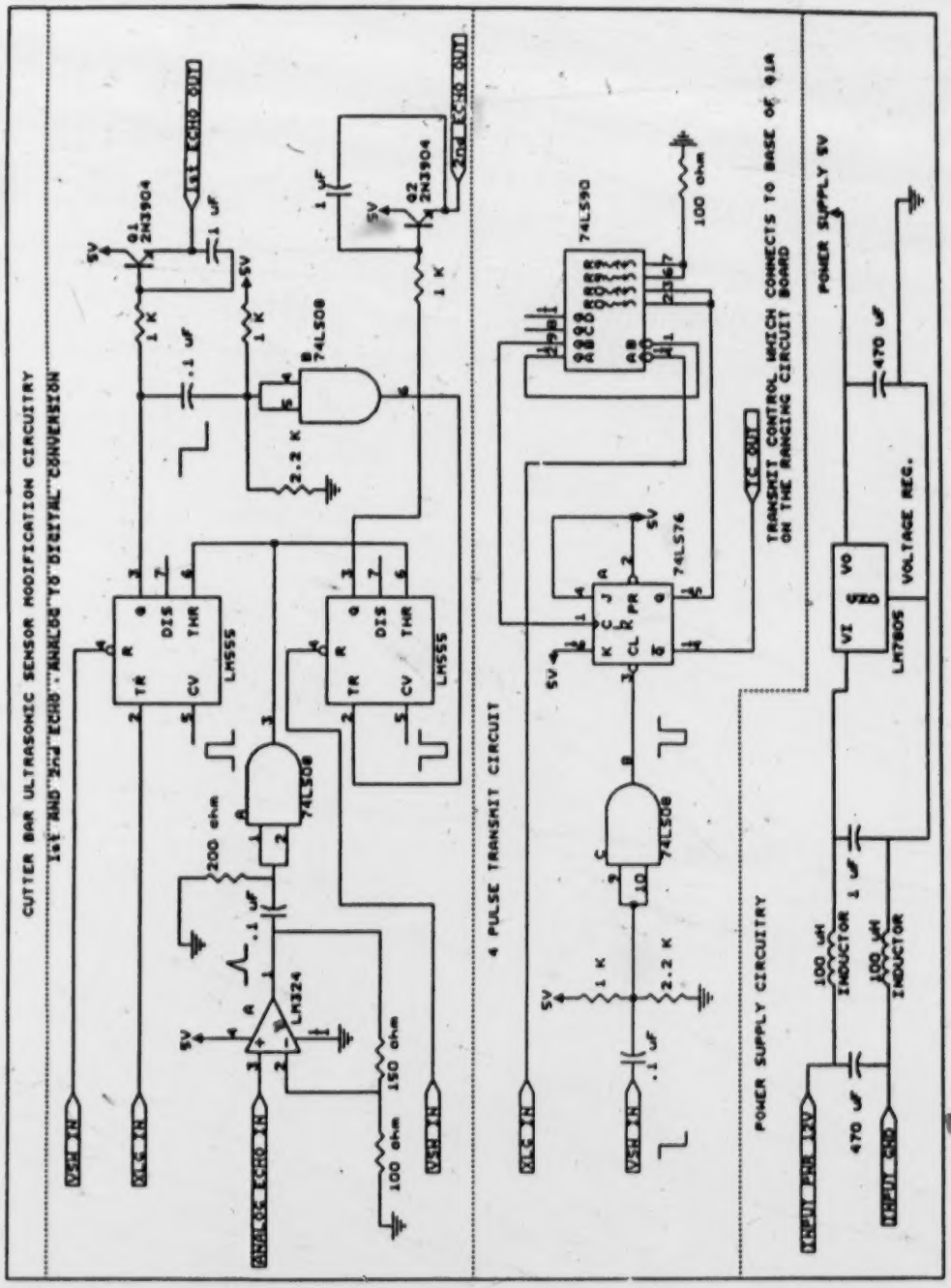


Figure 4.9 Cutter bar sensor modification circuitry

The two digital pulse duration signals were produced using two 555 timers, one operational amplifier, and two AND gates (Figure 4.9). The XLG signal from the ranging circuit board was connected to the trigger input of the first echo 555 timer, which produced a low-to-high transition on the output line when the transmit cycle began. The output of the first echo 555 timer remained high until a high pulse occurred at the threshold input, which indicated that the first echo had been detected.

The high pulse, which signified detection of the first echo, was produced from the processed echo analog signal displayed by pin 9 of U1 which was located on the ranging circuit board. This signal was amplified using an LM324 operational amplifier and a positive-edge-input/positive-pulse-output circuit.

Negative-edge-input/negative-pulse-output circuitry, triggered by the output signal of the first-echo 555 timer, was used to latch the second-echo 555 timer via the input trigger. The output remained high until a high pulse occurred at the threshold input, which indicated that a second echo had been detected. The threshold input signal was produced from pin 9 of U1 and associated circuitry.

First and second echo digital output signals were current-amplified using two 2N4401 NPN transistors (Q1 and Q2, Figure 4.9). Also, two modifications were made to the Polaroid ranging circuit board. Resistor R1 was replaced with a 0 to 20 kΩ potentiometer which allowed the detector gain to be adjusted. Pin 9 of U1 was shunted to ground with a 100 kΩ resistor to shorten the discharge time of capacitor C4. This modification enabled a target to be detected at a minimun distance of 30 mm from the end of the pipe or sonic reflector.

Printed circuit boards were made to fit into the sensor tubes (51 mm I.D. schedule 80 PVC pipe) and populated with the above components. Housing the electronic circuitry in the sensor tubes provided a weather proof shield and minimized the possibility of corruption of signals by noise.

4.3 Microprocessor

A microprocessor was utilized to perform tasks as scheduled by the operating system. The operating system is a collection of machine language programs that use the microprocessor's resources to acquire data, display information, and output control signals. The microprocessor performed the following operations:

- read status of operator control switches
- updated display console readout
- read crop height sensors
- interrogated the cutter bar-to-ground height sensors
- analyzed sensor data
- updated hydraulic solenoid control
- stored sensor information on a storage device

The system was housed in protective cabinets allowing portability and placement in a tractor cab.

4.3.1 Microprocessor Hardware

The controller was based on an 8 bit central processing unit, (MC6809, Motorola Semiconductor Products Inc., Austin, TX.) which was interfaced with various peripheral components. This processor was chosen because of availability and familiarity. The microprocessor system consisted of the following components:

- 3 parallel interface adaptors (PIA) used to receive and send digital information
- 1 programmable timer module (PTM) used to collect pulse width data
- 2048 bytes of random access memory (RAM) used to store temporary data
- 2048 bytes of read only memory (ROM) used for operating system software storage
- 2 asynchronous serial communication interface adaptors (ACIA) used for data acquisition during testing and evaluation

The microprocessor system is outlined in detail below.

4.3.1.1 Clock and reset

The clock circuit consisted of a 3.57 MHz oscillating crystal connected to ground through two 27 pF capacitors. The crystal was connected to the EXTAL and XTAL pins on the MC6809 central processing unit (CPU). The crystal was used to activate the CPU's internal oscillator at 1/4 of the crystal operating frequency. A reset circuit was implemented which reset the microprocessor during power up. This switch was debounced using two 7414 Schmitt trigger inverters. A 10 kΩ pull-up resistor was used to pull the reset high when operating (Figure 4.10).

4.3.1.2 Buffers

To prevent possible loading on the data, address, and control lines, one octal bus transceiver (74LS245) was used for data line buffers, and two octal buffer/line driver/line receivers (74LS244) were used for the address and control lines (Figure 4.10).

4.3.1.3 ROM and address decoding

Address lines A12 through A15 were decoded using a 74LS154 four line to sixteen line decoder in order to divide the addressable space into 4 K-byte blocks. Output from the decoder consisted of 16 select lines used to address one of 16 4 K address blocks. The top 4 K of addressable memory was fully decoded and used as ROM space. The 2716 erasable programmable read only memory (EPROM) would respond to an addressable range of F800 to FFFF (hexadecimal notation). Address space ranging from F000 to F7FF was unused but available as fully decoded ROM memory for future expansion. Full decoding was accomplished by using address line A11 and a NAND gate (Figure 4.11).

4.3.1.4 RAM

A high speed static RAM component (6116, 2048 X 8 bit) was placed at the address location D000-D7FF. The RAM component would respond to two different address ranges, D000-D7FF, and D800-DFFF, because the addressable field was not fully decoded (Figure 4.12).

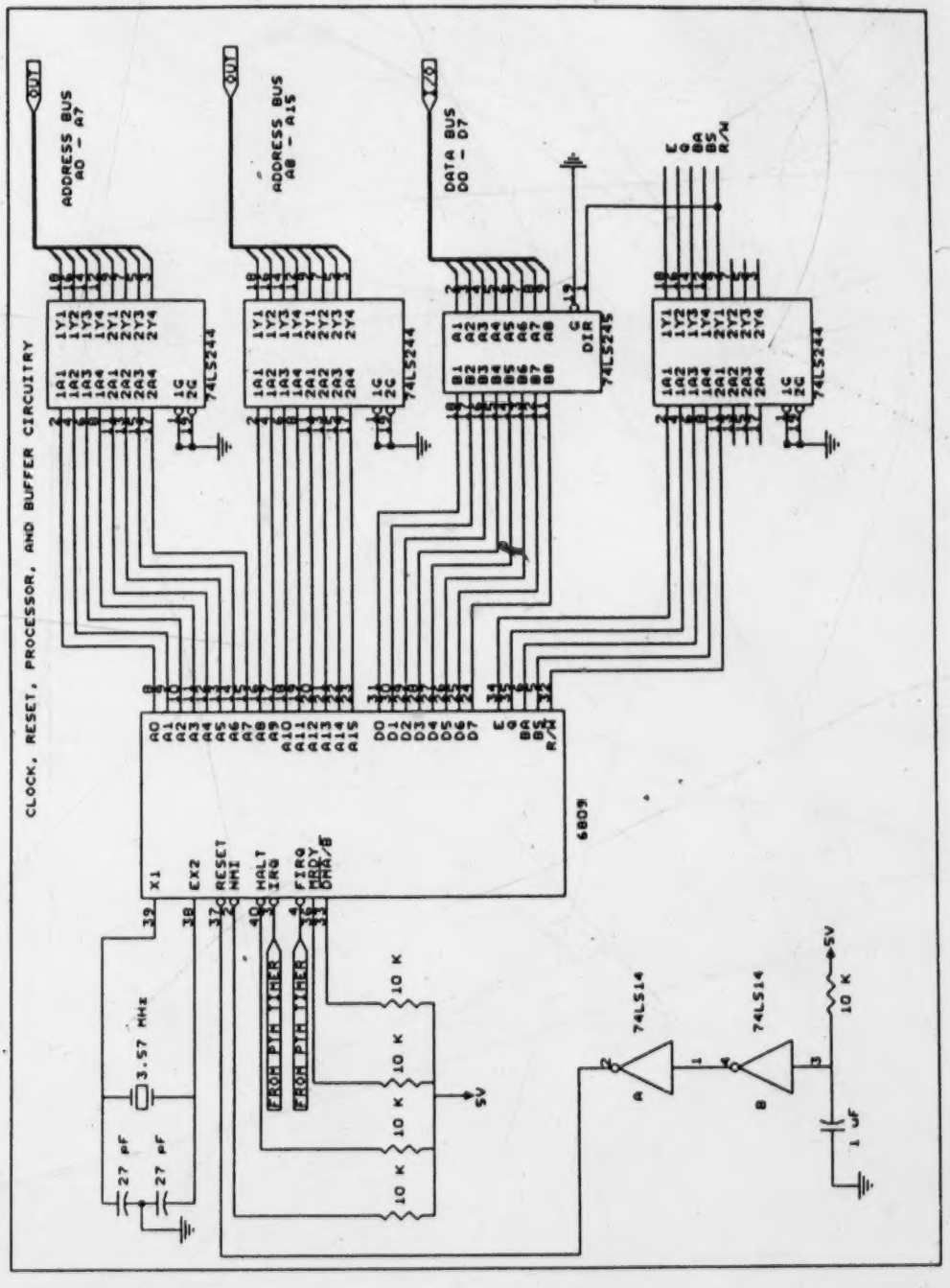


Figure 4.10 Microprocessor, clock, buffer, and reset circuitry

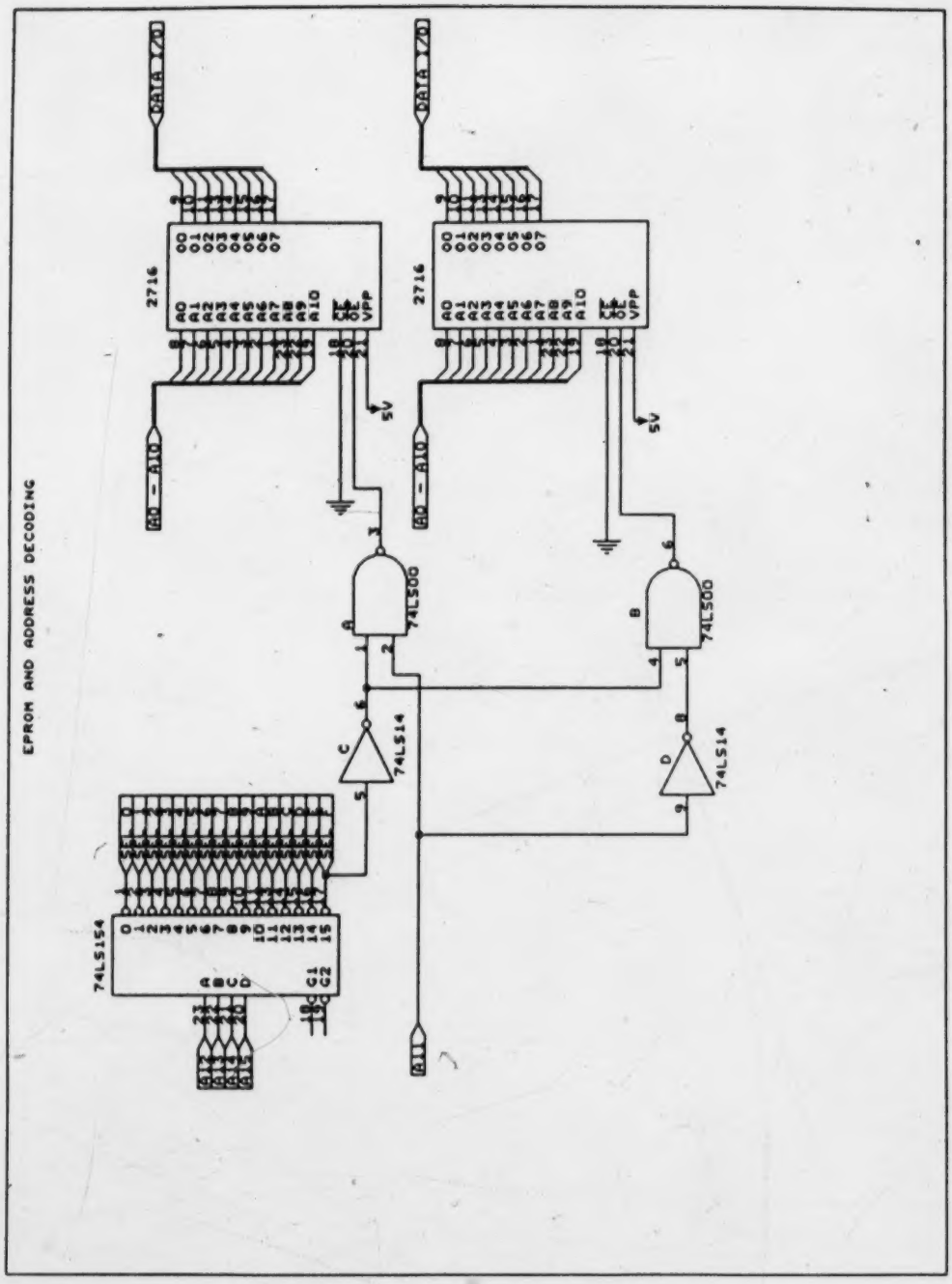


Figure 4.11 EPROM and address decoding circuitry

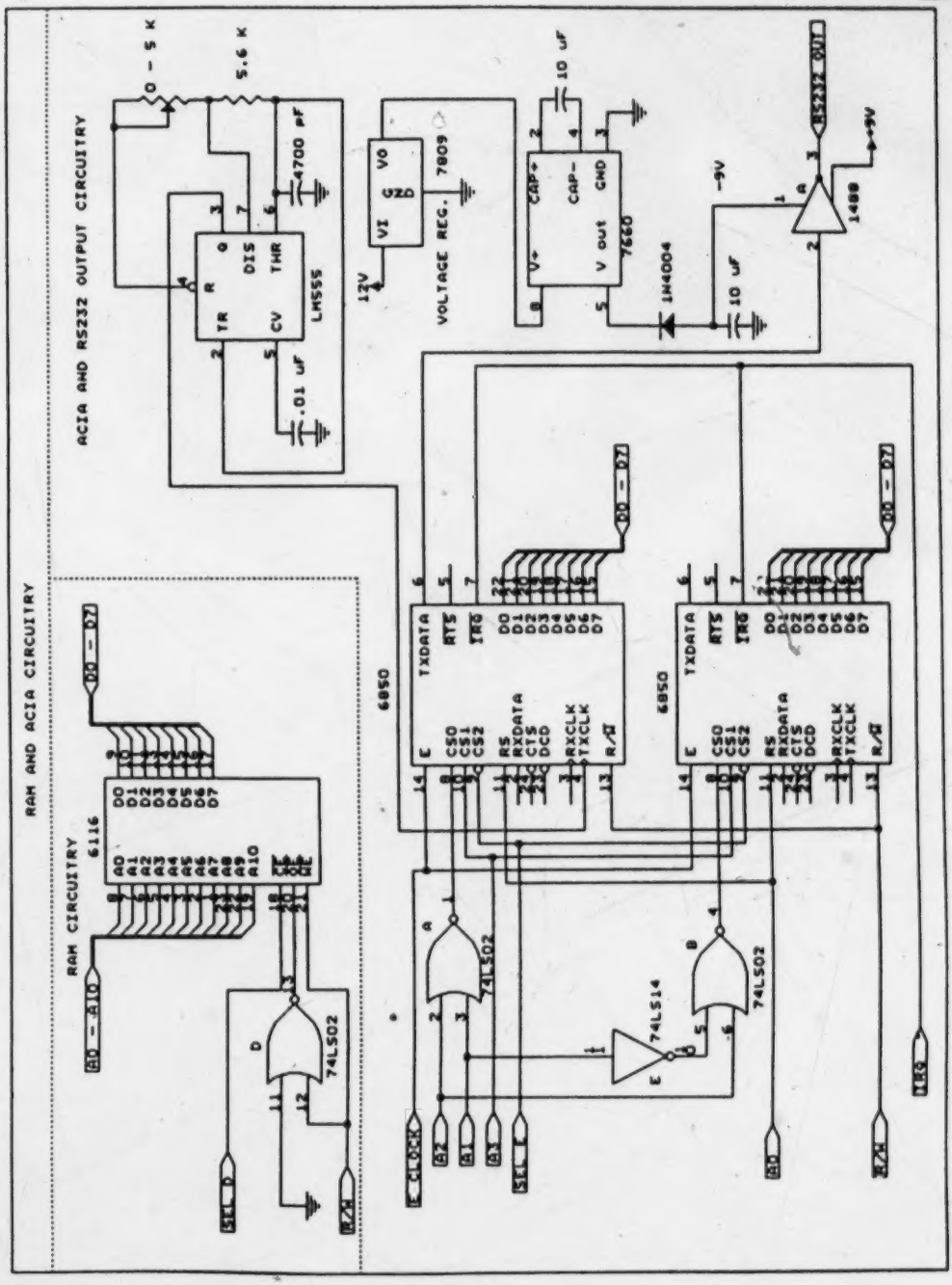


Figure 4.12 RAM and ACIA circuitry

4.3.1.5 ACIA

Two MC6850 ACIA's were connected to the basic system. One of these was used for writing information to a data storage device during testing and evaluation. The second ACIA was not used with the controller but was made available as an output port capable of sending data to a dumb terminal for design and testing of the controller hardware. One ACIA was located at address E008-E009, while the second ACIA was located at address E00A-E00B (Figure 4.12).

4.3.1.6 PIA

Three MC6821 PIA's were connected to the system to provide a means of parallel data communication for operator input switches, data acquisition from cutter bar and reel height sensors, and output control lines used to actuate windrower hydraulics. The first PIA, labelled PIASW (Figure 4.13), was used to read input switches manipulated by the operator in order to interact with the controller. This PIA was placed at address location 8000-8003. The remaining two PIA's were used both for crop height sensor inputs and for windrower hydraulic control outputs. Each peripheral data bus in each PIA was dedicated to one of four (A,B,C,D) sets of crop height sensor inputs and windrower hydraulic control outputs (Figures 4.14 and 4.15). PIA configuration and corresponding address locations are outlined in Table 4.2 below.

Table 4.2 PIA address locations

PIA label	Sensor set	Address
PIAA	A	C000-C001
PIAB	B	C010-C011
PIAC	C	C020-C021
PIAD	D	C030-C031

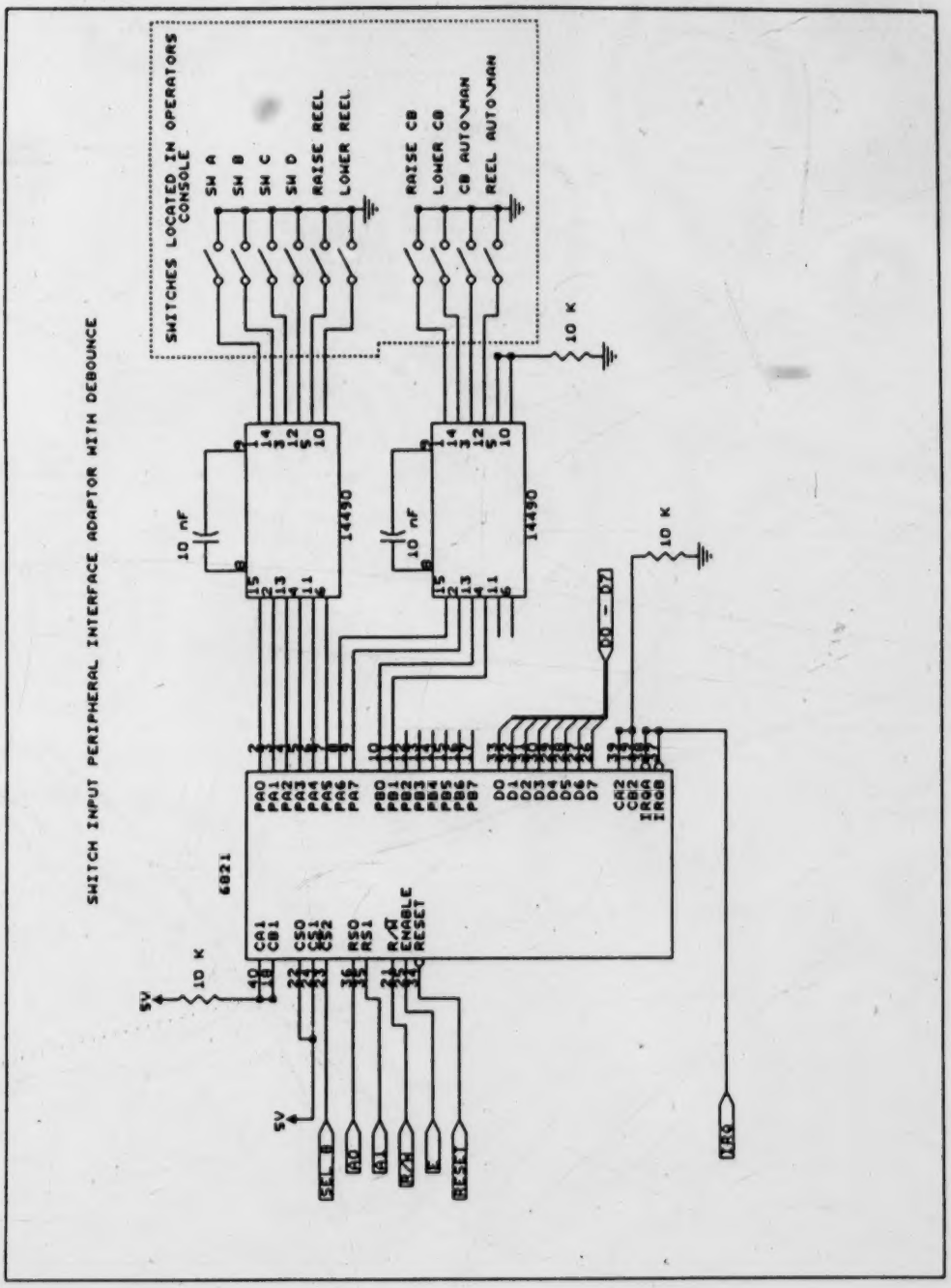


Figure 4.13 Switch input PIA and debounce circuitry

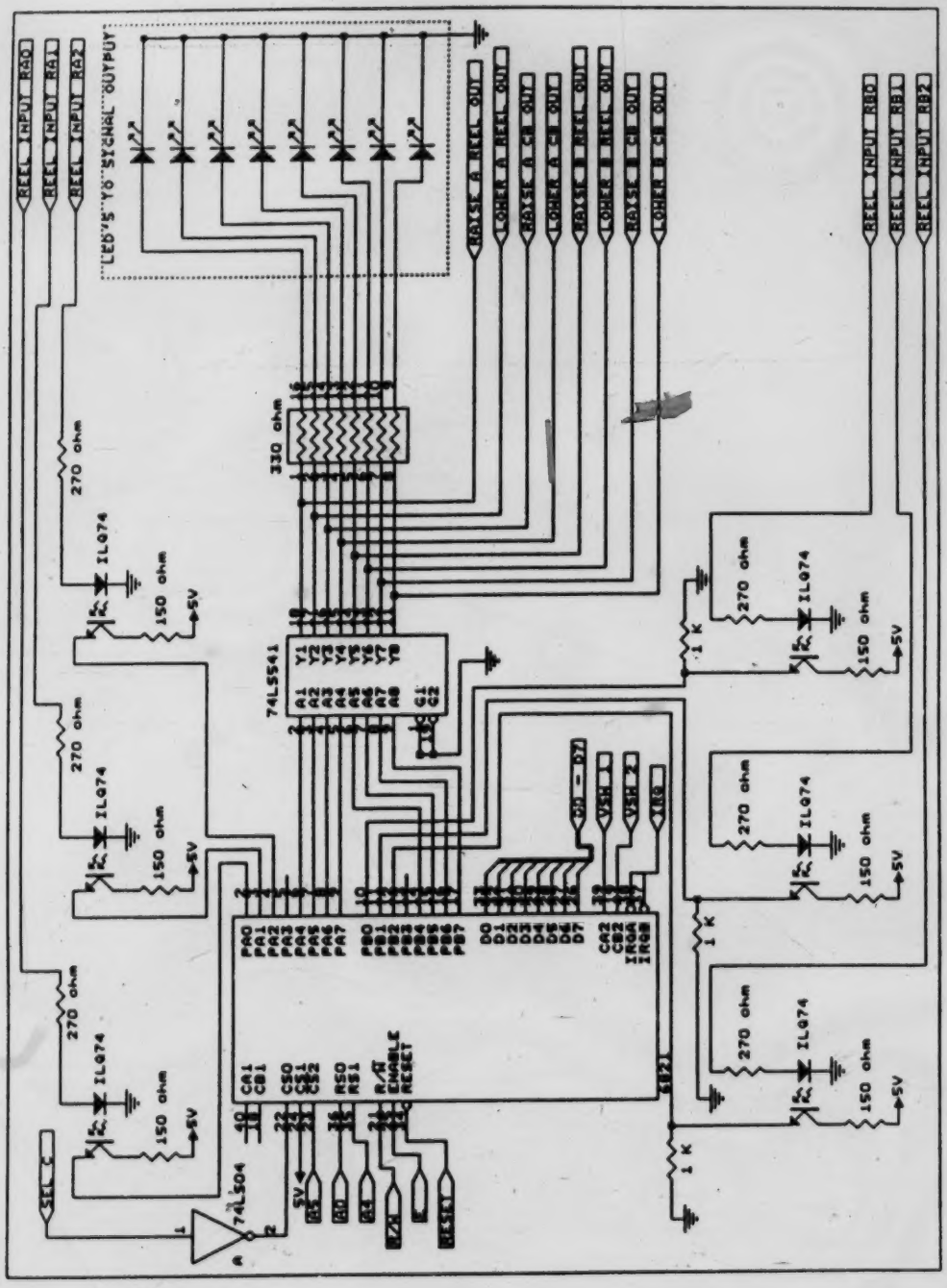


Figure 4.14 Sensor sets A and B PIA circuitry

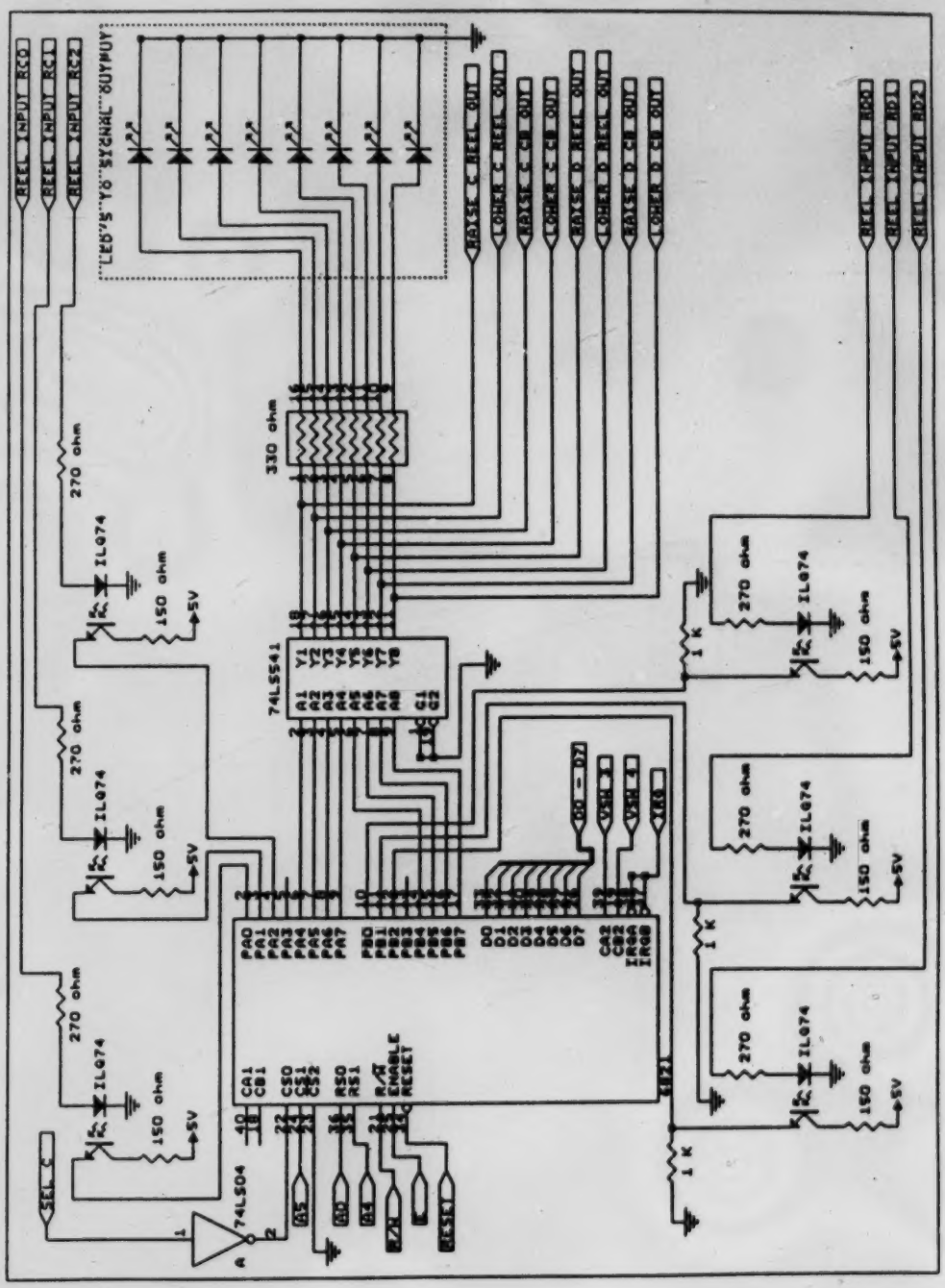


Figure 4.15 Sensor sets C and D PIA circuitry

4.3.1.7 PTM

An MC6840 PTM was incorporated into the basic system at address E000-E007. The PTM consisted of three programmable timers used for timing both internal and external events. Timers #2 and #3 were used to time external events associated with the cutter bar height sensor's first and second echo inputs. Timing began when a negative transition, caused by the echo inputs, occurred on the input gates G1 and G2 (Figure 4.16). Timers #2 and #3 were configured to produce a system interrupt when a second negative transition occurred on the input gates G1 and G2, corresponding to first and second echo returns, which originate from the cutter bar height sensor. Timer #1 was used as an internal counter to generate a system interrupt when sensor echos were not detected within a certain length of time. Figure 4.17 illustrates the PTM interface to the processor system.

4.3.1.8 Operator control switch inputs

An operator's console was used to allow operator interaction with the controller. Ten switch inputs were necessary for the operator to change control parameters. Each switch was debounced using two hex debounce components (MC14490). A 10 nF capacitor was used in conjunction with the MC14490 to produce a 100 Hz debounce frequency. The debounced switch signals acted as inputs to the PIASW component (Figure 4.13).

4.3.1.9 LCD

To display controller process information to the user, an alphanumeric liquid crystal display (LCD) module (Printed Circuits International PCIM 200) was mounted in the operator's console. This module allowed 16 dot-matrix characters to be displayed at one time. The character size was approximately 3 mm wide and 6 mm tall. An LCD display was chosen over a light emitting diode (LED) display because LED readability is impaired when exposed to direct sunlight, a condition which can occur in the windrowing environment.

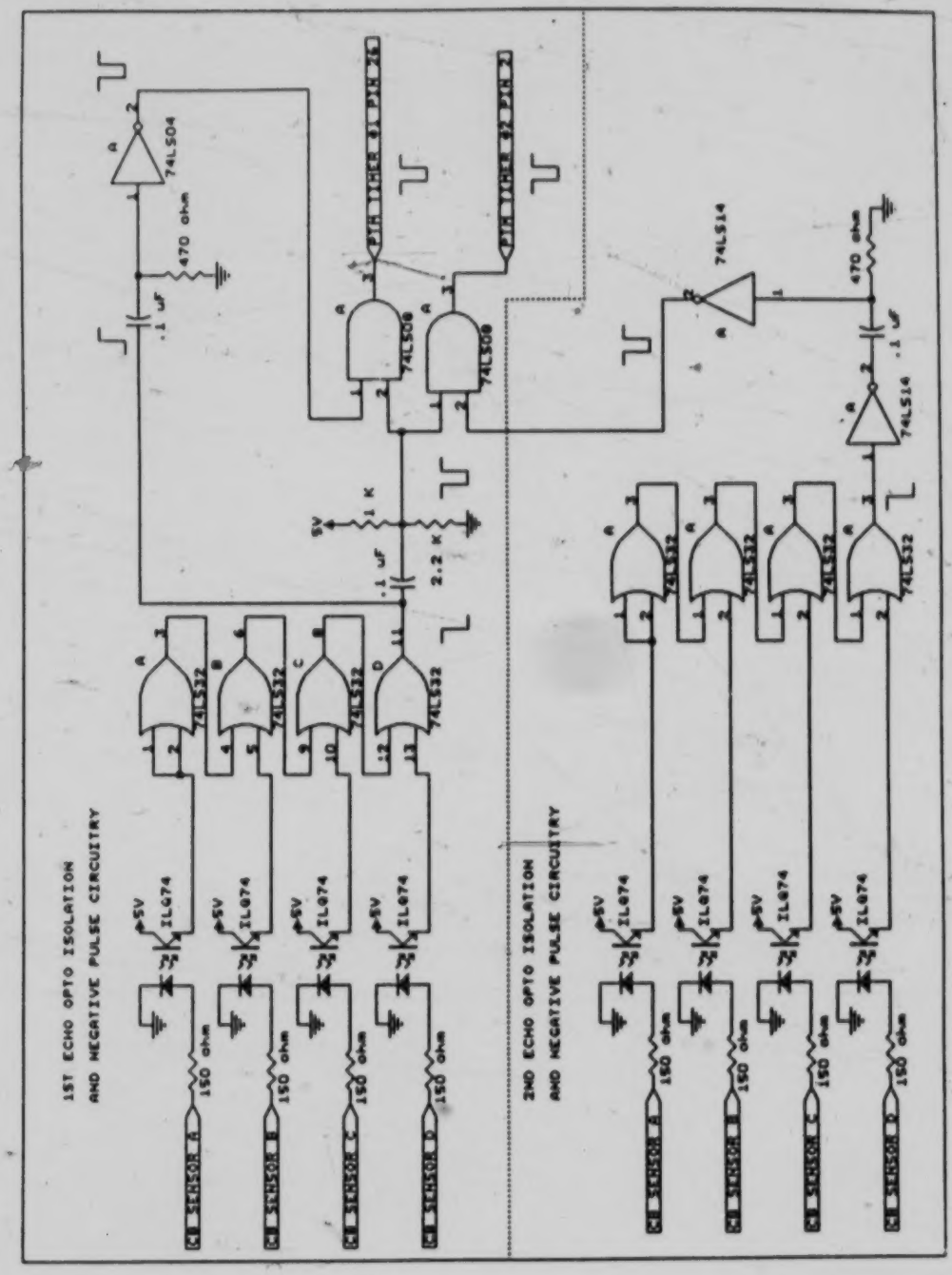


Figure 4.16 PTM gate driving circuitry

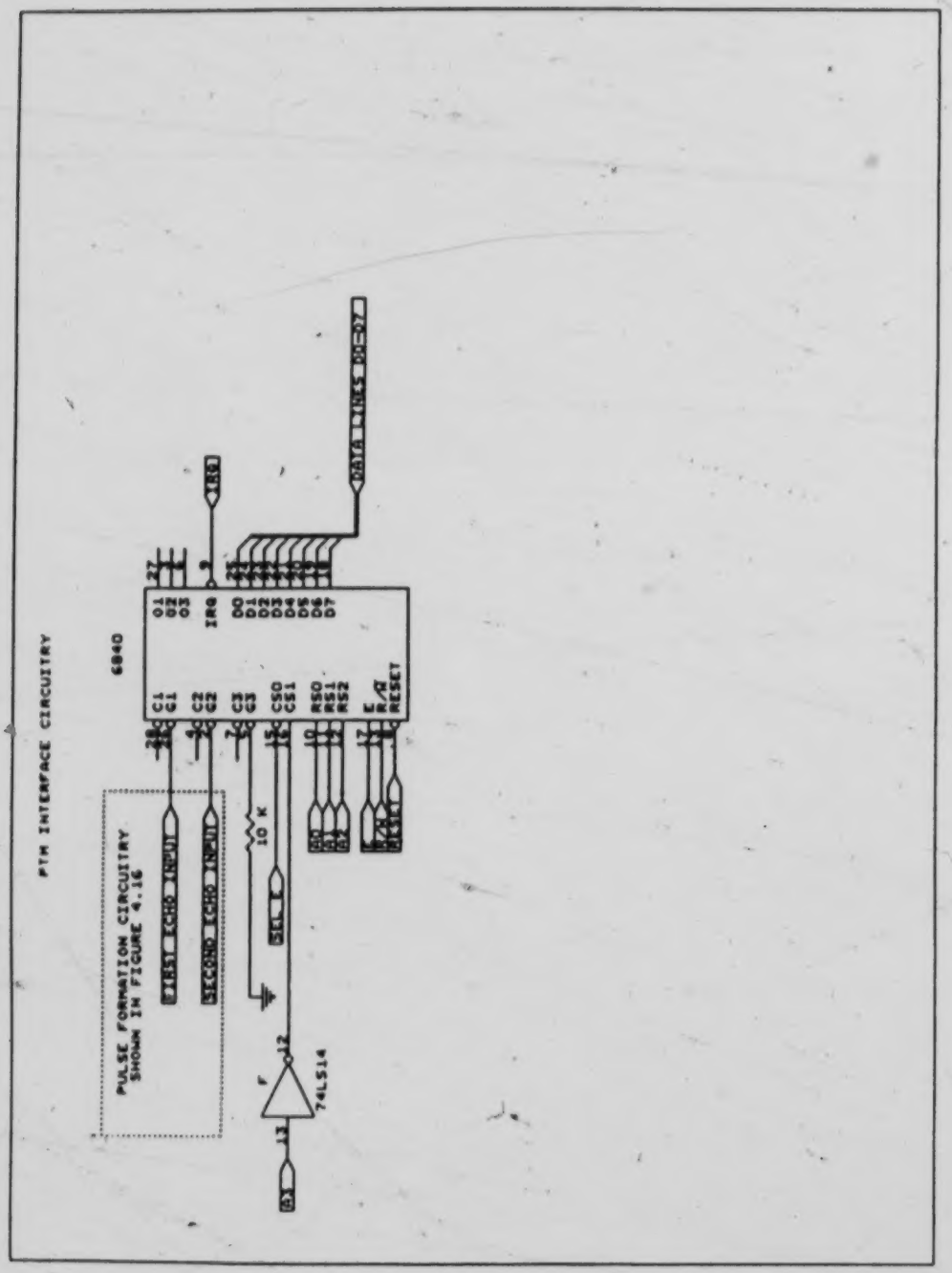


Figure 4.17 PTM interface circuitry

Thirteen lines were required for the LCD interface with the microprocessor. Eight data lines were connected to the microprocessor data bus using a bus transceiver (74LS244), which was enabled whenever data transmission was required. One active-low, chip-select pin was provided for address enable. The chip-select pin was connected to the SEL 0 address decoder output (74LS154), which allowed the LCD to be accessed at address locations 0000-0FFF, as the addressable range was not fully decoded. The read/write enable pins, MRD, and MWR were connected to the microprocessor read/write line. A 5V power supply and ground also were connected to the display.

4.3.1.10 Sensor input interface

Each crop height sensor had three digital data output lines, D0, D1, and D2, whose functions are described in Table 4.1. The data output lines from each crop height sensor were interfaced with the microprocessor through opto-isolation circuitry, which electrically isolated crop height sensor voltages from the microprocessor voltage system. An ILQ74 opto-coupler component was used. The LED of each coupler was driven by the crop height sensor data lines. The coupled photo-transistors drove the appropriate PIAA, PIAB, PIAC, and PIAD input lines high whenever a high voltage signal from one of the crop height sensors occurred (Figures 4.14 and 4.15). Cutter bar height sensor first-echo and second-echo data lines, as well as VWS control lines, were interfaced with the microprocessor utilizing ILQ74 opto-isolators (Figure 4.16).

4.3.1.11 Hydraulic control output interface

Sixteen control outputs were required to control four sets of reel heights and cutter bar heights. Control initiated from PIA output lines which were used to drive the ILQ74 opto-isolator LED's. The photo-transistors, which are coupled to the LED's, then drove the base of power Darlington transistors (2N6387) located in the interface junction box. The power transistors were used to energize solenoid-hydraulic valves (Figure 4.18).

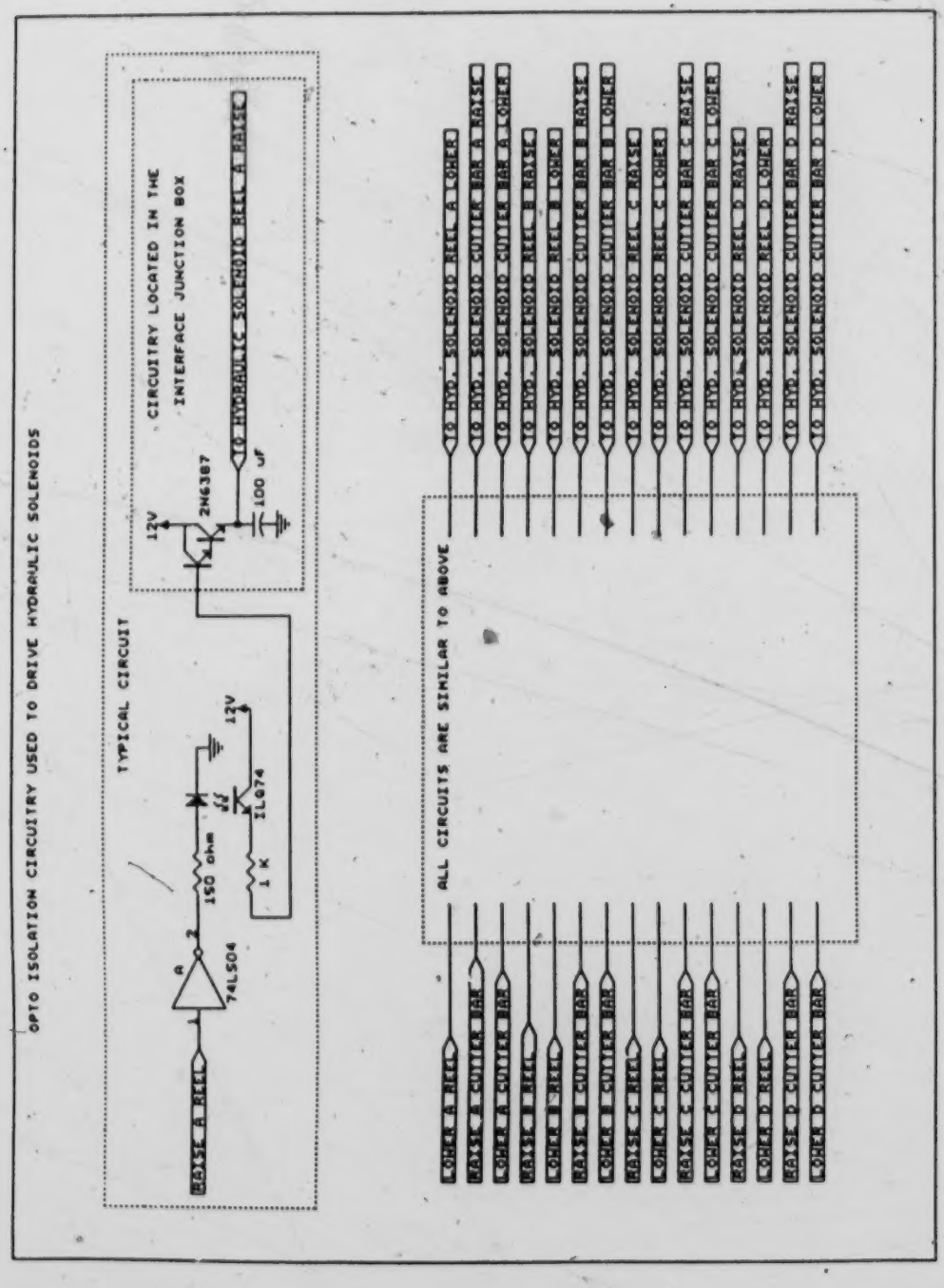

OPTO ISOLATION CIRCUITRY USED TO DRIVE HYDRAULIC SOLENOIDS
TYPICAL CIRCUIT
CIRCUITRY LOCATED IN THE INTERFACE JUNCTION BOX
12V
2N6387
TO HYDRAULIC SOLENOID REEL A RAISE
100 uF
12V
150 ohm
1 K
IL074
74LS04
RAISE A REEL
ALL CIRCUITS ARE SIMILAR TO ABOVE
TO HYD. SOLENOID REEL A LOWER
TO HYD. SOLENOID CUTTER BAR A RAISE
TO HYD. SOLENOID CUTTER BAR A LOWER
TO HYD. SOLENOID REEL B RAISE
TO HYD. SOLENOID REEL B LOWER
TO HYD. SOLENOID CUTTER BAR B RAISE
TO HYD. SOLENOID CUTTER BAR B LOWER
TO HYD. SOLENOID REEL C RAISE
TO HYD. SOLENOID REEL C LOWER
TO HYD. SOLENOID CUTTER BAR C RAISE
TO HYD. SOLENOID CUTTER BAR C LOWER
TO HYD. SOLENOID REEL D RAISE
TO HYD. SOLENOID REEL D LOWER
TO HYD. SOLENOID CUTTER BAR D RAISE
TO HYD. SOLENOID CUTTER BAR D LOWER
LOWER A REEL
RAISE A CUTTER BAR
LOWER A CUTTER BAR
RAISE B REEL
LOWER B REEL
RAISE B CUTTER BAR
LOWER B CUTTER BAR
RAISE C REEL
LOWER C REEL
RAISE C CUTTER BAR
LOWER C CUTTER BAR
RAISE D REEL
LOWER D REEL
RAISE D CUTTER BAR
LOWER D CUTTER BAR

Figure 4.18 Circuitry to drive hydraulic solenoids

4.3.1.12 RS232 communications interface

During field testing of the controller, sensor information was transmitted to a TRS-80 portable computer, where data could be stored permanently. One of the two available ACIA's was utilized to send the data to the TRS-80 using RS232 serial format.

Since outputs from the ACIA appear as TTL logic levels, these signals were converted to RS232 format utilizing an MC1488 RS232 line driver. An ICL7660 voltage converter was used to supply the MC1488 with a -9V power supply. A 555 timer was used in astable mode to provide a square wave at a frequency of 19.2 kHz to time serial data transmission at 1200 baud (bits/s). See Figure 4.12.

4.3.1.13 Power supply, connections, and cabinetry

All control equipment, including the microprocessor, was designed to run on a 12V power supply which, typically, is available on most farm equipment. Filtering of voltage irregularities was accomplished using a 4700 μF filtering capacitor along with a 5V, 3A voltage regulator. A 2A fuse and a 6A rectifying diode were installed for short circuit and reverse polarity protection. The microprocessor required 1.5A during operation. Power was taken from the tractor through the cigarette lighter socket. Figure 4.19 shows the power supply circuitry. The microprocessor and associated circuitry was enclosed in a metal cabinet to protect the circuitry from physical damage and electrical noise. A D-connector and ribbon cable was used for the 25 line connection between the microprocessor and the operator's console. A J-connector and a 40 conductor ribbon cable was used to connect the microprocessor with the interface junction box.

4.4 Operator's Console

The operator's console was used to interact with the machine control process and provide status information to the operator via a 16-character LCD (Plate 4.2). The number of control switches was minimized by using a menu to prompt the operator as parameters were changed. The operator's console provided the following functions:

- automatic or manual control of both the reel and cutter bar heights

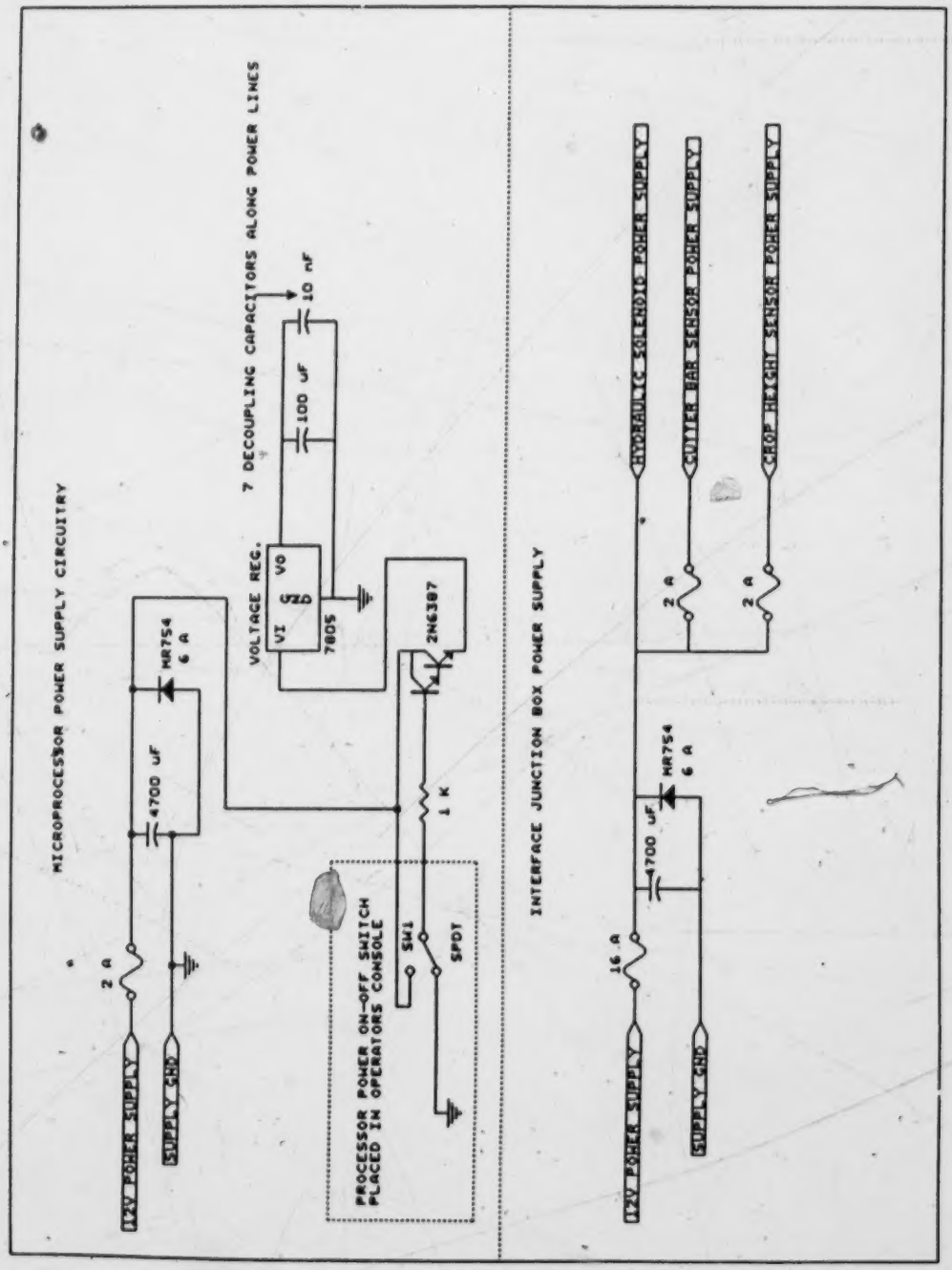


Figure 4.19 Power supply circuitry for the microprocessor and interface junction box

- constant display of 1 or 4 cutter bar heights
- automatic increment and decrement of cutter bar height set points

Four push-on push-off switches were used to select the control position (A,B,C, or D) corresponding to the set of hydraulic controls and sensors at which a setpoint was to be changed. When the four switches were off, all four positions were controlled together.

Two single-pole, double-throw, spring-return-to-normally-open switches were used to raise and lower the cutter bar and reel hydraulics or setpoints when manual or automatic modes, respectively, were selected. To select the mode of operation, two single-throw, single-pole switches were used to provide automatic or manual control: one switch for the cutter bar, and one switch for the reel.

Plate 4.2 Operator's console

4.4.1 Menu operation

Switches A through D were used to select the control position of the cutter bar and reel to be interrogated. If two switches were pressed, the position selected was that of the first switch pressed. When a position switch was selected, the cutter bar height for that position was displayed on the LCD along with the crop height sensor reading if the controller was in automatic mode.

With the cutter bar control switch in AUTO, and all selector switches open, the display showed:

AUT A## B## C## D##

where: **##** = actual cutter bar heights (in)

If a selector switch was pressed the display changed to the following, with X representing the selected location (either A, B, C, or D):

AUT X CB## RL#

where: **##** = actual cutter bar heights (in)

= reel position (1 through 3)

While in AUTO, the raise/lower cutter bar and reel switches allowed setpoints to be changed. When all selector switches were open and the cutter bar raise/lower switch was activated, all four cutter bar setpoints were displayed and incremented if raise was selected or decremented if lower was selected, displaying:

SET A## B## C## D##

where: **##** = cutter bar setpoints (in)

Pressing the raise/lower reel switch displayed the reel setpoints corresponding to one of three positions that the sensor could be controlled at. The display then showed:

SET A# B# C# D#

where: **#** = reel setpoint (1 through 3)

If a selector switch was pressed and a raise/lower switch activated, only the selected position X (A, B, C OR D) underwent setpoint changes. The display then showed:

SET X CB## RL#

where: ##,# = cutter bar and reel setpoints

In manual mode, the raise/lower cutter bar and reel switches raised or lowered the reel and cutter bar as controlled by the operator. With all position switches open, all positions were raised or lowered at the same time. In this case the display showed:

MAN A## B## C## D##

where: ## = actual cutter bar heights (in)

If an individual location X was selected, activation of the raise/lower switches for the cutter bar or reel raised or lowered the cutter bar or reel at that position only. The display then changed to:

MAN X CB##

where: ## = actual cutter bar height (in)

4.5 Interface Junction Box

The interface junction box provided a microprocessor-sensor and a microprocessor-hydraulic solenoid valve interface. A 40-conductor ribbon cable was used to relay information from the microprocessor to the junction box. The four crop height sensors, four cutter bar height sensors, and 16 hydraulic solenoid control lines also plugged into the interface junction box.

The interface junction box provided a fuse-protected, capacitor-filtered, power supply for all sensors and all hydraulic solenoids (Figure 4.19). As the microprocessor could not supply enough current to drive the solenoids, power Darlington transistors were connected in the junction box (Figure 4.18) to provide sufficient power to drive the solenoids.

5. CONTROLLER OPERATING SYSTEM SOFTWARE

The operating system scheduled a continuous loop of tasks which constantly updated data input and output. The operating system performed the following routines:

- initialized all system components
- polled the cutter bar and crop height sensors 16 times per second providing sensor input data used for control purposes
- read the operator's console switches to determine operating state changes
- refreshed the console information display
- processed the new sensor data
- updated control according to sensor data and console switch status
- initiated the sensor data acquisition cycle

Figure 5.1 shows a flow chart of the main program routine. Subroutines called by the main program routine will be discussed in detail. Appendix A gives a complete program listing of the operating software.

5.1 Subroutine INIT

Subroutine INIT initialized the LCD, the three PIA's, the PTM, and the ACIA to operate in the required modes. Cutter bar setpoints were initialized at a value of 250 mm while the reel height setpoints were initialized at a value of one meaning that the crop height would be controlled between the two lowest piezo strips. This routine was called directly from the main routine.

5.2 Subroutine POLCR

This routine initiated one data acquisition cycle for each of the four crop and cutter bar height sensors. A data acquisition cycle was defined as one sensor reading of each sensor. Data for each group of crop and cutter bar sensors were stored in separate RAM locations, as depicted in the memory map shown in Figure 5.2. Crop height sensor data were read directly from the corresponding PIA's, which were coupled to the crop height sensor data lines. Cutter

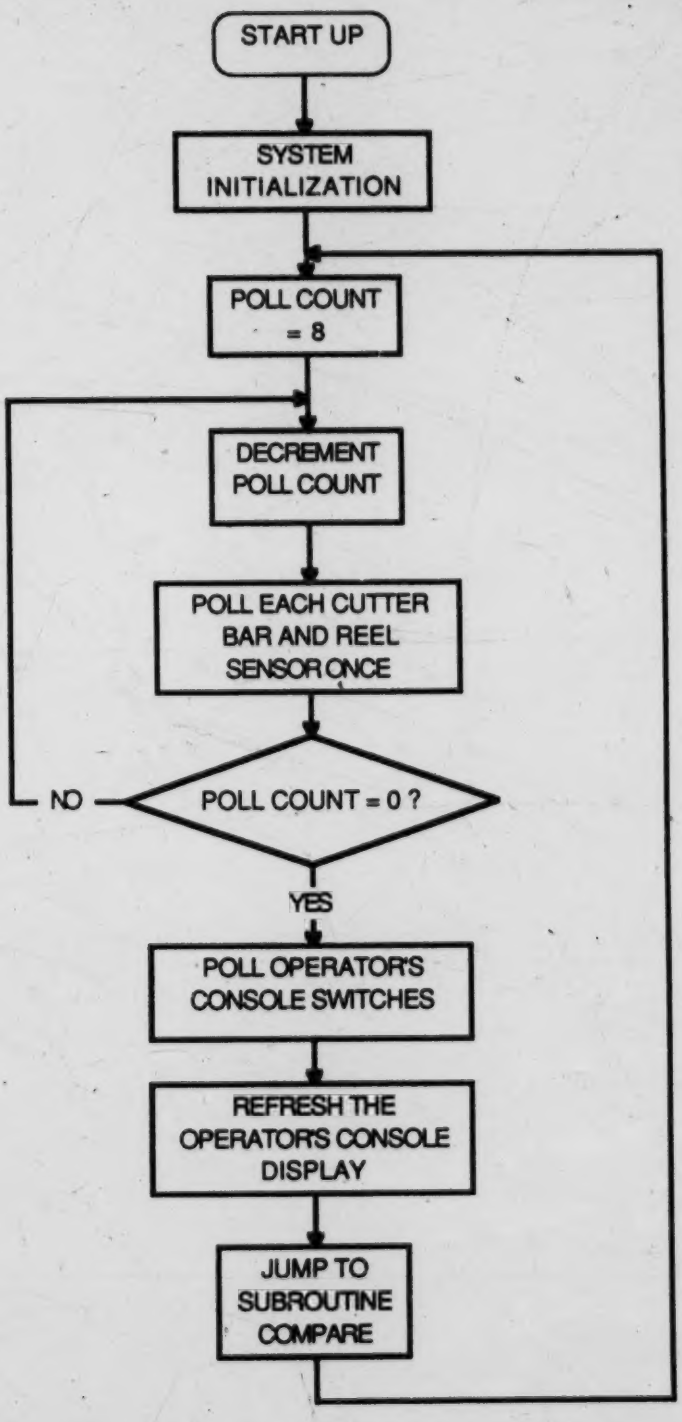


Figure 5.1 Main program routine flow chart

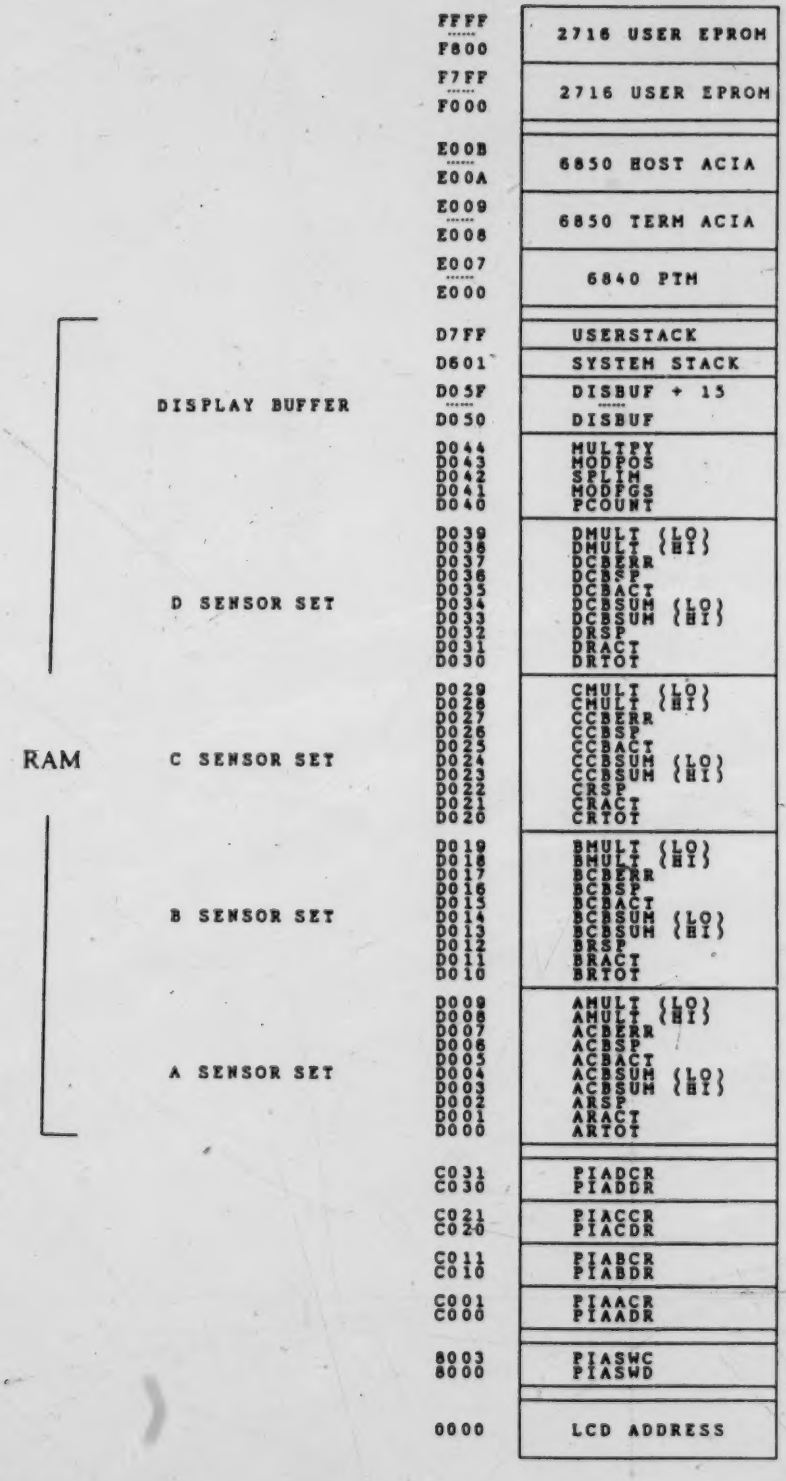


Figure 5.2 Controller memory map

bar height data were accessed by first setting the sensor VSW line through the appropriate PIA. This initiated sensor data acquisition. First and second echo data pulses were timed by the PTM and serviced in the FIRQ and IRQ interrupt subroutines as described below. After cutter bar height data had been captured, the VSW line for each of the four sensors was reset for the next polling sequence. When all sensors had been interrogated, the program returned to the main routine.

5.2.1 Subroutine FIRQ

This routine determined which PTM timer, #1 or #2, had initiated the interrupt. If timer #2 caused the interrupt, the first echo had been received and the count in the timer #2 PTM register was stored in RAM for later evaluation. If timer #1 caused the interrupt, the first echo was not recognized and an error condition had occurred. Timer #1 generated an interrupt if 14 ms elapsed without a first or second echo being recognized. The elapsed time period of 14 ms corresponded to a sensor-to target distance of 2400 mm, which was assumed to be beyond the limit of the sensor range.

When an error condition occurred, three tasks had to be performed. First, all timers were reset for the next sensor read sequence. Second, a no-echo flag was set to inhibit the POLCR subroutine from being executed, because a second echo would not exist if a first echo was not present. Third, an error count was updated which kept count of the number of errors received during eight data acquisition cycles. The error count was used to calculate the sensor-to-target distance based on the number of valid readings. The FIRQ routine returned to the POLCR subroutine upon completion.

5.2.2 Subroutine IRQ

This routine determined which PTM timer, #1 or #3, had caused the interrupt. If timer #3 caused the interrupt, the second echo had been received, and the count in timer #3 PTM register was stored in RAM for later evaluation.

If timer #1 caused the interrupt, the second echo was not returned indicating an error condition. Thus, the error count was updated.

Regardless of which timer had interrupted, all timers were reset for the next sensor read sequence, before the routine was terminated and returned to the POLCR subroutine.

5.3 Digital Display Routines

The digital display routines read the operator's console switches using PIASW (Figure 4.13) and configured a display buffer with the appropriate ASCII characters required for LCD operation.

Subroutines AUTOMAN and ABCD read the operator's console switches and updated a switch status register called MODFGS, which was configured as illustrated in Figure 5.3. The status of the reel auto-manual switch was stored in the REELAM status register.

Subroutine ADJUST read the cutter bar and reel raise lower switches and stored the result in the MODFGS register. Subroutine DISPLY stored the proper ASCII values in register DISBUF, which was used for LCD update in subroutine RFRSH. The DISPLY subroutine also changed cutter bar, reel height, and setpoint binary data into ASCII values for use by the

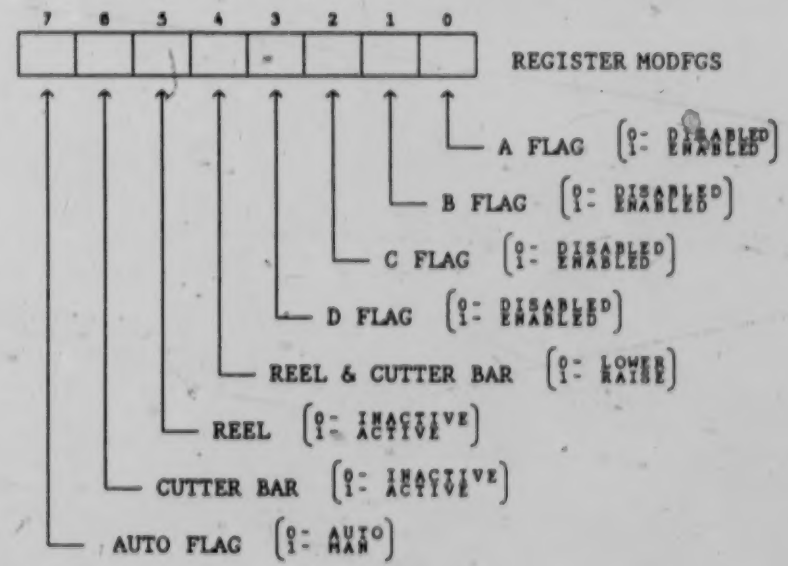

7 6 5 4 3 2 1 0
REGISTER MODFGS
A FLAG (0: DISABLED 1: ENABLED)
B FLAG (0: DISABLED 1: ENABLED)
C FLAG (0: DISABLED 1: ENABLED)
D FLAG (0: DISABLED 1: ENABLED)
REEL & CUTTER BAR (0: LOWER 1: RAISE)
REEL (0: INACTIVE 1: ACTIVE)
CUTTER BAR (0: INACTIVE 1: ACTIVE)
AUTO FLAG (0: AUTO 1: MAN)

Figure 5.3 Configuration of the switch status register MODFGS

LCD display.

Subroutine RFRSH loaded the updated DISBUF register, which contained 16 ASCII characters, into the LCD. The digital display routines were called after eight data acquisition cycles, as was scheduled by the main program routine.

5.4 Data Processing and Control Routines

Control and setpoint update of both the reel and cutter bar were performed by several subroutines in this section of software. Operation of the program is outlined in the flow chart as shown in Figure 5.4. Subroutine COMPAR was called by the main routine and determined whether auto or manual control of the cutter bar was selected. The processor was then directed to subroutine AUTO or subroutine MANUAL as determined by the operator's switch setting. Subroutine AUTO computed each crop and cutter bar sensor height, checked for errors, updated setpoint changes, and initiated control of the reel and cutter bar as required.

Crop and cutter bar sensor data was collected by the POLCR subroutine, and was acquired eight times before the height positions were calculated. The collected data was summed and stored in a sum total register. Processing of this information yielded an average sensor reading based on the past eight data acquisitions with provisions made for error handling. Cutter bar height sensor errors occur when an echo is not received by the sensor. Crop height sensor errors occur when the crop detection microswitch does not detect crop at the cutter bar.

The crop height sensor position was determined by dividing the sum total register by eight. Valid sensor data corresponded to a binary number 0 through 4 indicating the number of piezo strips in contact with the crop. An average number greater than 4, indicated an error condition existed, because crop was not present at the cutter bar which was indicated by the crop detection microswitch. This information was passed from the crop height sensor to the microprocessor in a binary 7 count form (Table 4.1). The error condition was flagged to the operator by placing a '?' where valid data was usually displayed on the LCD. During an error condition, the reel was controlled manually as illustrated in the flow chart Figure 5.4.

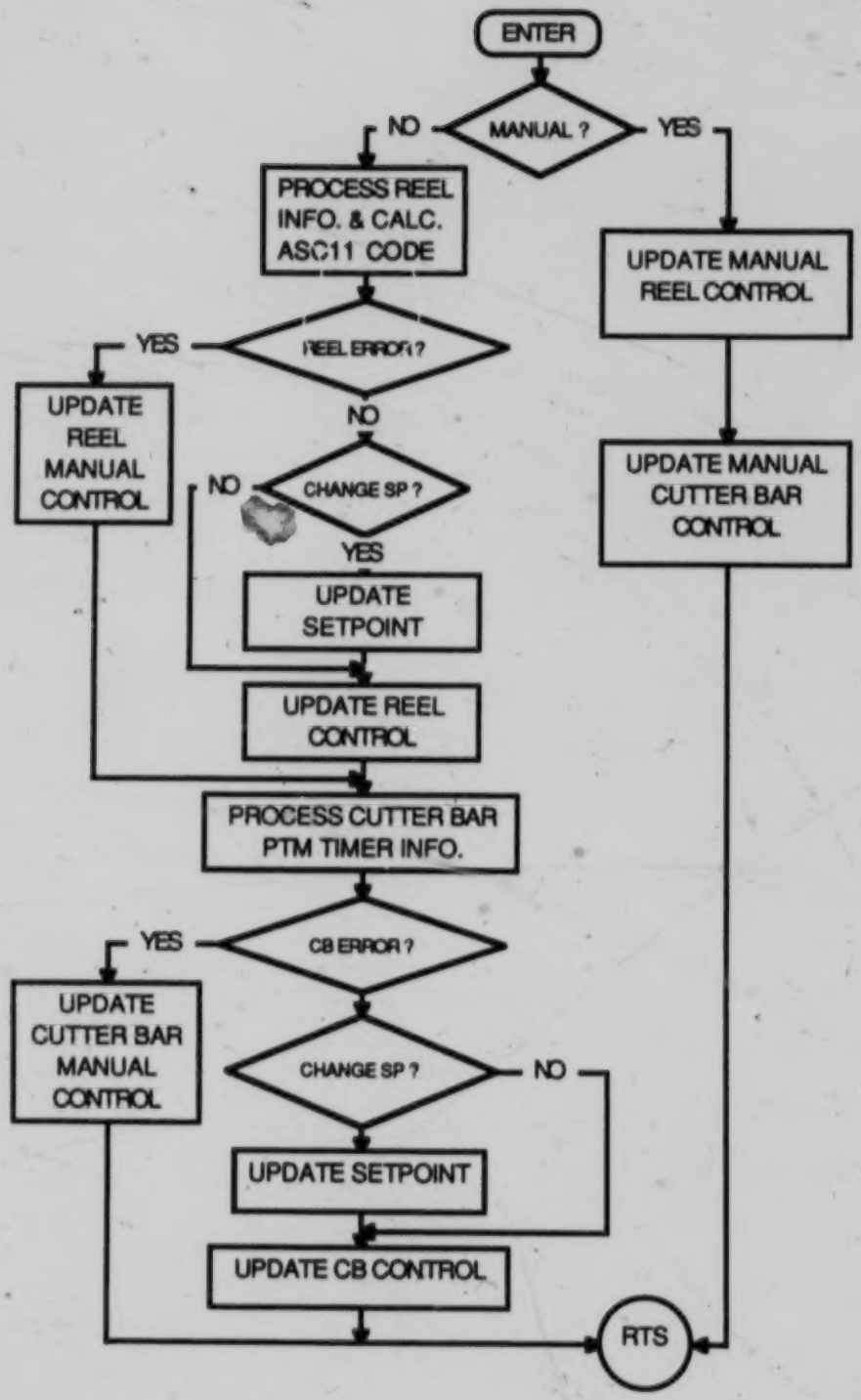


Figure 5.4 Data processing and control routine flow chart

The temperature-compensated cutter bar height was computed using the following formula:

$$DT = [[(P_1/16 + P_2/16)/(P_1/256) * 10]/16 \cdot 10] * D_1 \cdot OFF \qquad 5.1$$

where: DT = distance to the target

P_1 = time elapsed from transmit to first echo received (ms)

P_2 = time elapsed from first echo received to second echo detect (ms)

D_1 = distance from transceiver to first echo target (mm)

OFF = offset used to zero sensor (mm)

The above formula made use of divide by 16 and 256 to make division of binary numbers simple, minimize rounding errors, and maintain a 16-bit word length for accuracy. P_1 and P_2 held the sum of eight timing counts produced by the PTM for first and second echos respectively. D_1 represented the distance from the ultrasonic transducer to the end of the sensor tube or first echo target. OFF is the offset used to set the distance from the end of the sensor tube, where the sensor would read zero. This allowed the sensor to be positioned above the cutter bar inhibiting contact with the ground while displaying proper height readings. This equation was independent of the number of samples acquired as the calculation was performed on a ratio basis. Thus, a valid height was calculated as long as one out of eight second echos were detected while each sensor went through the polling sequence.

An error condition, corresponding to no second echo detected, occurs only if eight consecutive readings failed to detect a second target echo. An error condition was flagged by displaying '??' on the LCD where a valid height number would be displayed. During an error condition, automatic control was aborted, and manual control of the cutter bar was initiated.

If the cutter bar and reel raise-lower switches were activated by the operator during automatic control, the system setpoints were updated. Setpoints were updated according to the operator's console switches in subroutine SPCHNG.

When the crop height sensor was error free, control update of the reel position was initiated in subroutine RCMPAR. Three position on-off control was implemented as on-off, solenoid-hydraulic control valves were used to raise and lower the reel. The setpoint and

actual crop height sensor values were compared to determine if the reel should be raised, lowered, or remain unchanged according to the following requirements:

(SP-ACT ≥ 1) LOWER REEL

(SP = ACT) REMAIN IDLE

(ACT-SP ≤ 1) RAISE REEL

where: SP = set point value

ACT = actual crop height sensor reading

Control update of the cutter bar height occurred in subroutine CBCMPR when valid data was available. Three position, on-off control was initiated due to the limitations of the solenoid-hydraulic control valves used during field testing. The cutter bar height setpoints and sensor values were compared and updated according to the following strategy:

(SP-ACT) > 50 (mm) RAISE CUTTER BAR

|SP-ACT| ≤ 50 (mm) REMAIN IDLE

(SP-ACT) < 50 (mm) LOWER CUTTER BAR

where: SP = set point value

ACT = actual cutter bar height sensor reading

Subroutine MANUAL processed sensor data information, and updated manual control of the reel and cutter bar as required. The cutter bar and reel data calculations were accomplished by executing the same subroutines used in the subroutine AUTO. Error conditions were displayed in the same format as described in the AUTO subroutine description above. This information was for display purposes only, as control was updated by the operator.

Manual control of both the reel and the cutter bar was initiated by reading the status of the cutter bar and reel raise-lower switches on the operator's console, and activating the

proper output lines accordingly.

5.5 Data Capture Routine

During testing, provision was made to allow cutter bar height sensor data to be transmitted to an external data storage device. Subroutine SEND was called during the POLCR subroutine, while the microprocessor was waiting for a return echo signal from the cutter bar height sensor. ASCII characters representing the cutter bar sensor heights were sent through the ACIA in asynchronous serial data format at 1200 baud to a target system. The ACIA was configured in the INIT subroutine for seven data bits, even parity, and two stop bits.

6. TESTING

Field testing of the complete control system and sensors was conducted during the 1987 fall harvest season in both wheat and barley crops, and over wheat stubble, summerfallow, and hay land terrain conditions.

Windrowing, while using the controller, was carried out on a 20 ha wheat field for cutter bar height control analysis, and on a 5 ha barley field for reel height control analysis. The wheat and barley fields were located near D'arcy, Saskatchewan.

Cutter bar height sensors were tested further for height reading repeatability while travelling over wheat stubble, sumerfallow, and hay land terrain conditions. Testing was performed at the University of Alberta's Ellerslie Research Station in Edmonton, Alberta.

6.1 Materials and Methods

A 7.3 m-wide pull-type windrower (Model 35, Massey Ferguson), was used for cutter bar and reel height control testing. Two Parker solenoid-activated, three-position hydraulic valves were used: one to control the reel height, and one to control the cutter bar height. The two valves were mounted on the back of a Model 1486 International tractor, which was used to pull the windrower during testing.

The microprocessor box, interface junction box, and operator's console were all placed inside the tractor cab. The microprocessor was placed behind the seat, the interface junction box placed on the floor, and the operator's console placed on the left side of the steering wheel. A portable computer (TRS-80 Radio Shack) was used to collect data from the microprocessor and was placed behind the tractor seat, when required. Plate 6.1 illustrates the component placement. The tractor 12V electrical system was used to power the control microprocessor, sensors, and hydraulic solenoids. Power was taken from the cigarette lighter socket on the tractor.

For cutter bar sensor height evaluation over wheat stubble, summerfallow, and hay terrain conditions, two sensors were mounted on the drawbar of a Model 2805 Massey Ferguson tractor. Mounting the sensors on the drawbar of the tractor allowed for a constant

sensor-to-target distance to be maintained during testing. Again, to collect data from the microprocessor during testing, a portable computer (TRS-80 Radio Shack) was used. The microprocessor, portable computer, interface junction box, and operator's console were all placed in the tractor cab.

6.1.1 Cutter bar control testing methods

The cutter bar height control was evaluated in a wheat field with a heavy infestation of Russian thistle. Crop stand was very light having a grain yield of 0.65 t/ha. Ground terrain was hilly and greatly undulating having 15 % grade slopes. These extreme conditions were ideal for cutter bar height controller evaluation because of the constant terrain height variations, and poor ground visibility, which are characteristic of the environment that a controller must operate in.

Two cutter bar height sensors were evaluated: one utilizing the sonic reflector, placed horizontally in front of the cutter bar, and one without the sonic reflector, placed vertically on the side of the windrower. Plate 6.2 depicts the sensor orientations and mounting. Both sensors were positioned 150 mm off the ground surface, measured with the cutter bar lowered onto the ground. This offset was required to prevent the sensors, particularly the horizontal one, from coming into contact with the ground. The two cutter bar height sensor locations and configurations were used to determine if one orientation had an advantage over the other.

Two control software strategies were initiated in order to study their effect on control stability. First, the eight sensor readings, which were averaged for control update, were increased to 16 sensor readings. When the 16 sensor readings were used for averaging, control output to the solenoid valves was pulsed on-off at a frequency of 1 Hz. Pulsing only occurred when control of the solenoid valve was required. This control technique will be referred to as "pulse-action". Second, when eight sensor readings were averaged for control update, no pulsing action took place. This control technique will be referred to as "no-pulse action".

Four windrowing passes were executed for stubble height measurement purposes. Two sensor orientations, horizontal with the sonic reflector, and vertical without the sonic

Plate 6.1 Component placement in tractor cab

Plate 6.2 Cutter bar and reel sensor orientations and mountings

reflector, and the two software control subroutines, pulse action, and no pulse action were utilized in a 2 X 2 experimental design in order to study control effectiveness. Only one replication of each treatment could be carried out due to field and time limitations.

Evaluation was carried out between two steep hills having 15 % slopes by measuring the terrain slope and stubble heights after the crop was cut. Measurements were taken every 1 m along the windrower path with the use of a surveying level and a range pole. A windrowing speed of 7.5 km/h and a stubble height setpoint of 250 mm were held constant during control evaluation.

Control performance of the cutter bar height was recorded on a video tape as an aid for visual evaluation.

6.1.2 Reel control testing methods

Reel height control performance was evaluated in a field of barley, which had been damaged by early frost. The light bearded heads proved to be an excellent testing crop. A strong wind, which blew during the testing period, enhanced the severity of test conditions, as wind had an effect on crop-sensor impacts. If the windrower travels in the direction of the wind, crop-sensor impacts will be lighter than if the windrower was travelling against the wind. Thus travelling with the wind and, hence, minimizing crop-sensor impacts, provided a rigorous test procedure. The barley field terrain was flat, which minimized disturbances in reel height due to terrain elevation differentials. As a further step to minimize disturbances in the reel height other than crop canopy variation, the cutter bar height control was placed in manual mode and not adjusted.

The crop height sensor was mounted on an extension of one reel support arm. This placed the sensor in front of the reel so that the sensor was in contact with the undisturbed crop material. The probe of the sensor was offset 15-20° off vertical and angled towards the windrower. This orientation was used in an attempt to aid in crop height detection. Placing the sensor at such an angle allowed the crop material to contact the sensor without interference from the lower part of the sensor. Plate 6.2 shows the placement of the sensor.

The method of evaluation began by lowering the reel fully before entering the barley crop. This was accomplished by placing the reel controller in automatic mode, and disconnecting the crop detection switch, which caused the reel to lower fully when no crop was present. The windrower then was put into operation and driven into the crop, while the ability of the crop sensor to follow the top of the crop canopy was monitored visually. The windrower was driven into the crop with the wind as this was the worst case for testing. Reel height control performance was recorded on video tape to aid visual evaluation.

One modification to the reel height control software was made which enhanced control stability. The eight sensor readings, which were averaged for control update, were increased to 16 sensor readings. Control output of the reel height was then pulsed on-off at 1 Hz during reel hydraulic movement.

Reel hydraulic vertical response was slowed from 300 mm/s down to 100 mm/s for control stability purposes. The slower hydraulic response was required as the crop height sensor was set to update crop height changes once a second.

6.1.3 Cutter bar sensor testing methods

The cutter bar height sensors were further evaluated for height reading repeatability over stubble, hay land, and summerfallow terrain surfaces, as such analysis could not be determined from the cutter bar height control testing. Wheat stubble heights varied between 100 and 150 mm. Residual straw and chaff was plentiful, so bare ground exposer was minimal. Hay land terrain consisted of a 1 m wide strip of alfalfa forage, which had stalk heights of 200-400 mm. Summerfallow was freshly cultivated with 50-75 mm diameter clods of surface soil. All terrain areas were flat with zero slope.

Two cutter bar height sensors were mounted on the drawbar of the Model 2805 Massey Ferguson tractor, which was used as a test vehicle. Drawbar mounting allowed the sensors to be maintained at a fixed distance above the ground surface, a requirement for the repeatability study.

One sensor was mounted horizontally utilizing the sonic reflector, while the other sensor was placed vertically with the reflector removed. Sensors were placed close together to allow detection of the same target area. Interference caused by the sound waves produced from both sensors was eliminated by polling each sensor with the other one inoperative. Mounting brackets allowed the height of both sensors to be varied during testing. Plate 6.3 illustrates the sensor positions and mountings.

The wheat stubble and hay land terrains were utilized in a 2 X 2 X 3 X 3 experimental design. Treatments consisted of the two sensor types, vertical without the sonic reflector, and horizontal with the sonic reflector, the two terrain types, three sensor to target heights 130, 380, and 740 mm, and three tractor speeds 5, 6.8, and 12 km/h. Three replications of each treatment were carried out. The experimental design was used to determine how travel speed, terrain type, sensor-to-target distance, and sensor orientation affected sensor accuracy and repeatability.

The three sensor-to-target heights were set when the tractor was placed on a smooth concrete surface. This was accomplished by moving the sensors vertically, until the proper height was displayed on the operator's console.

The two test plots for wheat stubble and hay land were 50 m in length and wide enough to be straddled by the tractor. Straddling the test plot ensured that the terrain read by the sensors was not trampled by the tractor wheels. Tractor speed was calculated by recording the time to travel the length of the test plot.

Testing over summerfallow terrain was carried out to determine the sensors ability to recognize a step change in elevation. The step change in ground elevation was simulated using a plywood board elevated 150 mm above the summerfallow surface. The summerfallow terrain test plot measured 25 m in length. Tractor speed was calculated to be 7.2 km/h by recording the time to travel the length of the test plot. A sensor-to-target height of 380 mm was used in this test. Also, the two sensor types, vertical without the sonic reflector, and horizontal with the sonic reflector, were evaluated over the summerfallow terrain. Three replications of this treatment were carried out.

Plate 6.3 Horizontal and vertical mounting of the cutter bar sensor

Data for each replication were recorded on magnetic tape (utilizing the TRS-80 portable computer) and then transferred into ASCII files on an IBM microcomputer. A Pascal program was written to analyze the raw data from each replication, and tabulate the results. Four statistics were calculated for each run: the mean sensor-to-target distance reading, a coefficient of variation, the number of errors encountered (an error condition occurred whenever a return echo was not detected), and the number of valid readings from which the mean and coefficient of variation were calculated. An analysis of variance (ANOVA) then was conducted using the means of the replications.

7. TESTING RESULTS AND DISCUSSION

7.1 Reliability

The controller hardware performed reliably with only minor malfunctions occurring while testing. The microprocessor unit gave problems associated with the power supply circuitry. After several hours of operation, the 5V, 3A voltage regulator failed, and required replacement. Electrical interference caused by the tractor's electrical system would cause microprocessor malfunction when the tractor was started with the microprocessor turned on. This problem was eliminated by turning on the microprocessor while the tractor's engine was running.

The interface junction box circuitry sustained damage because of inadequate protection from voltage spikes having 150 Vpp amplitudes, which were produced from the collapsing magnetic field in the solenoid-hydraulic valves when their power supply was disconnected. This problem was rectified by placing a 100 μF capacitor from ground to the emitter of the power transistor which dampened the damaging voltage spikes to 50 Vpp.

The cutter bar height sensor performance was unaffected by moisture, vibration, and dust. However, when the sensor was operated in the horizontal position, crop material would catch on the sonic filter in the end of the tube or attach to the sonic reflector and cause a false sensor reading. This problem occurred twice during the testing period, and was not considered to be of much concern. This problem could be rectified by providing better mechanical protection from the crop material.

The crop height sensor performed as required with one malfunction. After a few hours of operation, the on-board voltage regulator broke off at its mounting pins as a result of excessive vibratory movement. This problem was rectified by positioning the voltage regulator against the printed circuit board so that the regulator was restrained from movement.

7.2 Cutter Bar Height Control

Visual observations of the cutter bar height control showed unacceptable control as the stubble height varied ±100 mm from the setpoint value of 250 mm. Results obtained were similar for both sensor orientations and both "pulse" and "no-pulse" control algorithms. Unstable control was caused by a number of factors.

First, the control algorithm used three position, on-off control having a dead zone of ±50 mm around the setpoint. When the "no-pulse" software algorithm was used, control of the cutter bar height was updated after eight height readings were acquired and averaged. In real time, the cutter bar height was updated once every 0.5 s. The windrower hydraulic system raised or lowered the cutter bar vertically at a rate of 200-300 mm/s or 100-150 mm every 0.5 s. Unstable control resulted as a tolerance of ± 50 mm was used in conjunction with fast acting hydraulics. Quick movement of the cutter bar caused it to pass through the dead zone before the next update when correcting for control error. The "pulse" software algorithm eliminated this problem and corrected for some of the control instability.

Second, the horizontally mounted sensor utilizing the sonic reflector read the ground height 500 mm in front of the cutter bar. The sensor readings taken at this position were more sensitive to cutter bar movement because the added distance from the pivot wheel increased the speed of vertical movement. This increased the possibility of missing the dead zone when control was being updated.

Third, the mechanical design of the pull-type windrower produced a torsional spring action on the end of the cutter bar which gave rise to a damped vertical oscillation where the sensors were located. This oscillation was excited whenever the cutter bar height hydraulics were used. Thus, the sensor readings were affected by the amplitude of the oscillation at the instant when a sensor reading was taken.

Fourth, the heavy infestation of weeds sometimes inhibited the sensor from obtaining a valid reading from the ground as ground visibility was impaired by the weed canopy. One of two false sensor readings took place when the sensor was unable to sense the ground surface. The sensor could receive an echo from the top of the weed canopy, mistaking this reading for

the ground to cutter bar height, with the result that the cutter bar was raised too high. This occurred more often when green weeds were read by the sensor, as green leafy weeds tended to reflect sound more readily than dead Russian thistle.

The other false sensor reading occurred when the sensor failed to detect an echo. This would happen when the thick mat of Russian thistle absorbed the sound waves, and was particularly evident after a light snowfall. In this case the automatic control defaulted to manual control until a valid height reading was detected.

Figures 7.1 through 7.4 illustrate the results of the four windrowing passes which depict a graphical representation of cutter bar height control when windrowing between two steep hills. Figure 7.1 illustrates control with the horizontal sensor using no-pulse-action control; Figure 7.2 vertical sensor using no-pulse-action; Figure 7.3 horizontal sensor using pulse-action; and Figure 7.4 vertical sensor using pulse-action. The graphs show similar control, each graph depicting the same general trend. Each graph illustrates that stubble heights were equal to or less than the setpoint with the exception at the bottom of the hill where stubble height exceeded the setpoint. The setpoint was exceeded at the bottom of the hill because at that point, the tractor began climbing the opposite slope and raised the hitch, causing a disturbance in the cutter bar height. Stubble heights below the setpoint indicate that error conditions (sensor not receiving a target echo) aborted control by placing the controller in manual control mode. Error conditions occurred after the cutter bar was lowered, thus stubble height was cut short until a valid sensor reading was available to correct for the error. Another observation made from this data indicates that false echos originating from the top of the weed canopy did not occur while sensing over dead Russian thistle. This can be seen by the absence of stubble heights exceeding the setpoint height.

As all four figures show similar cutter bar control, no generalizations can be made on control performance as a function of sensor orientation or control algorithm.

Overall performance was inadequate, because variation in stubble heights were greater than the dead zone tolerance used in the controller software.

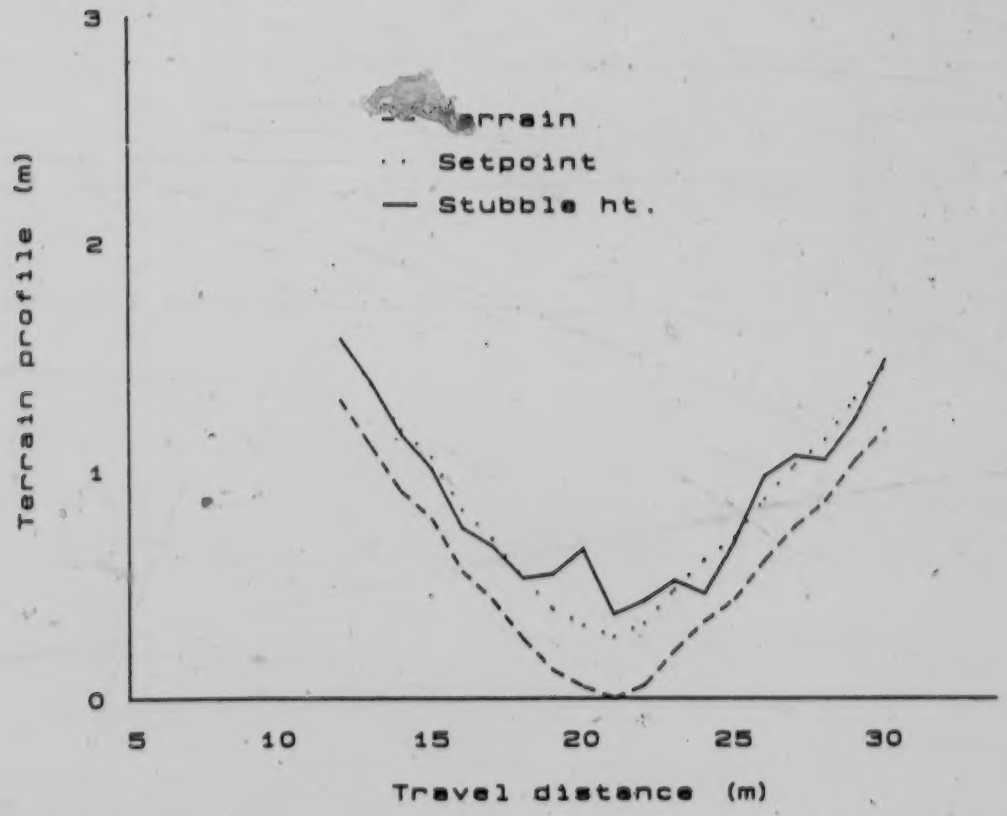


Figure 7.1 Cutter bar height control using the horizontal sensor and no-pulse control

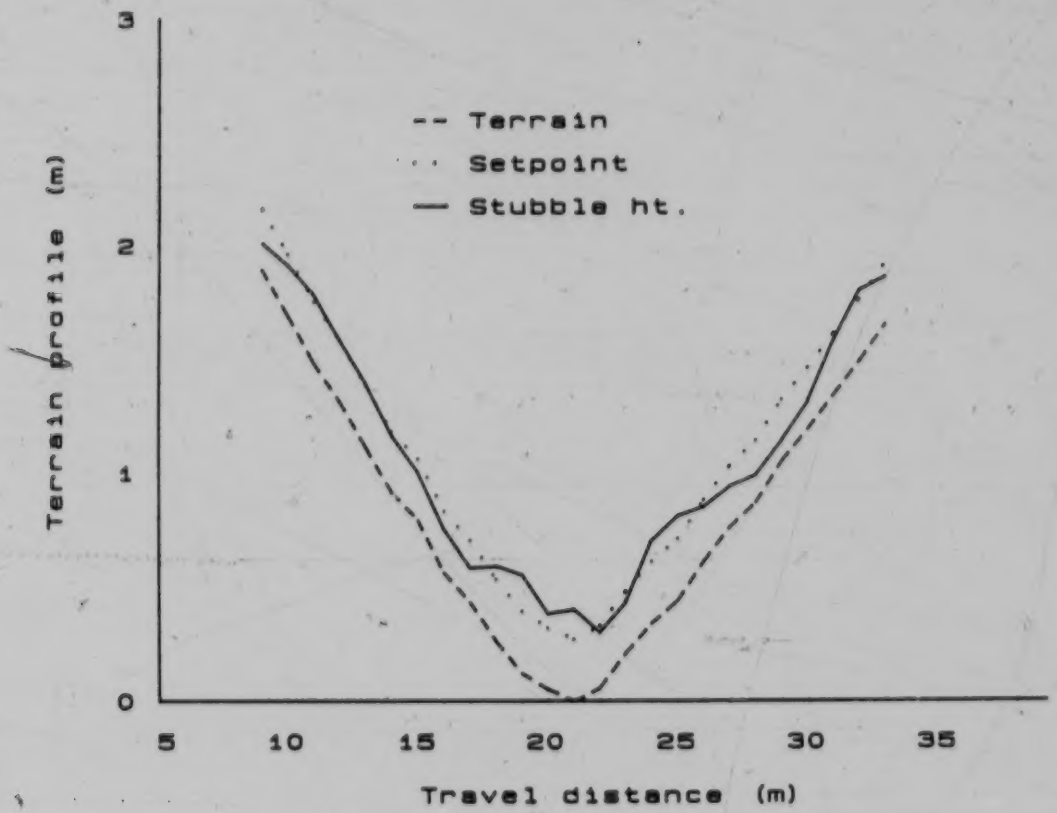


Figure 7.2 Cutter bar height control using the vertical sensor and no-pulse control

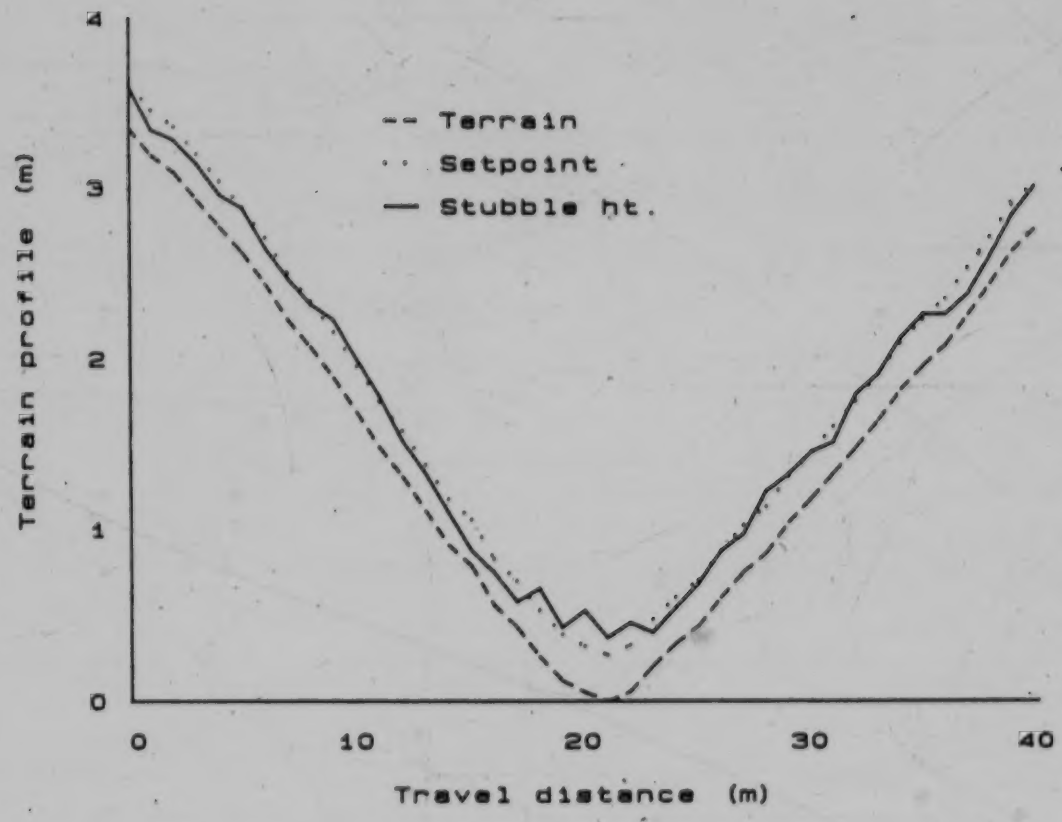


Figure 7.3 Cutter bar height control using the horizontal sensor and pulse control

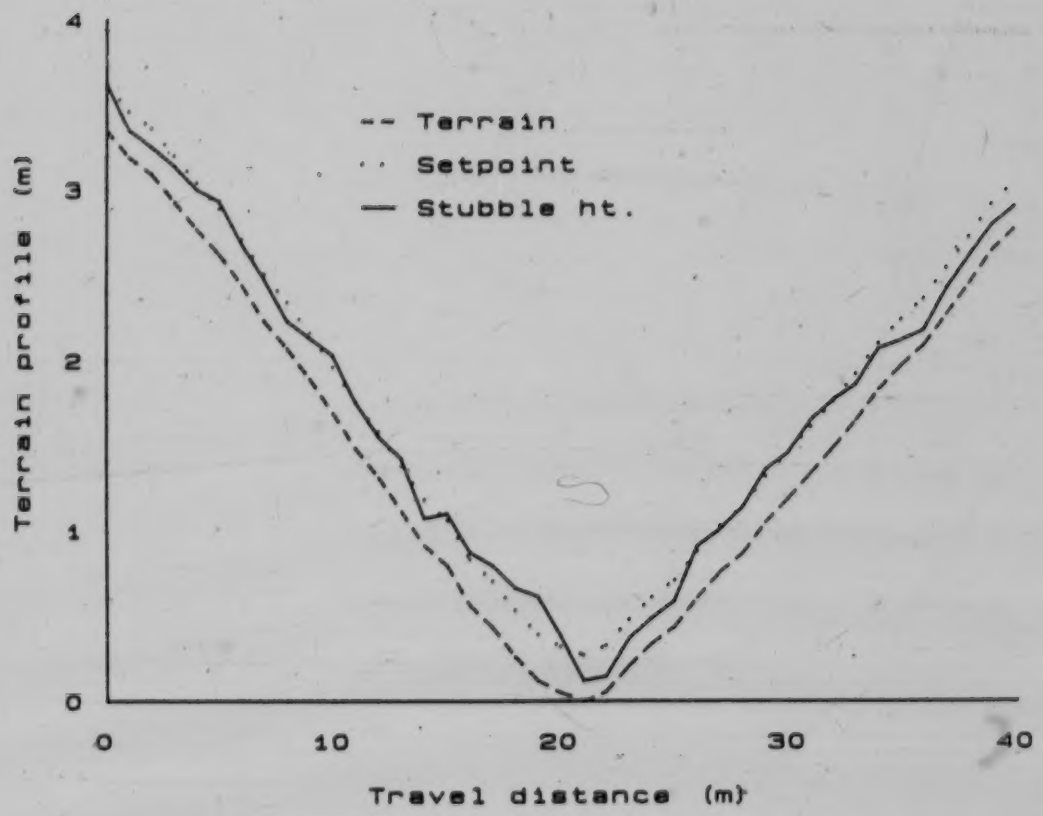


Figure 7.4 Cutter bar height control using the vertical sensor and pulse control

7.3 Reel Height Control

Best control occurred when the highest of the three height settings was chosen. The highest setting on the crop height sensor corresponded to the sensor setting where the crop canopy top was controlled between the lowest two piezo sensor strips. The other two settings were not effective for control as the sensor cabinet, which housed the electronic circuitry, would deflect the crop away from the piezo strips, inhibiting crop sensor contact when the reel was too low.

Generally, the reel would automatically raise to the proper position. The proper height was reached within 4 to 6 s or 8-12 m into the crop. However, if the crop was exceptionally thin, the crop height sensor had trouble sensing enough impacts to determine the crop height position.

7.4 Cutter Bar Height Sensor Evaluation

An analysis of variance (ANOVA) was conducted using the means of the test runs which are presented in Appendix B. The results of the ANOVA are presented in Table 7.1. Sensor type, terrain, and height effects showed significant differences at $p = 1\ \%$. Of the first order interactions, sensor x terrain was significant at $p = 1\ \%$, while sensor x height proved significant at $p = 2.5\ \%$. From these results several generalizations can be made on sensor performance and these are outlined below.

First, the ANOVA indicated a highly significant difference ($p = 1\ \%$) between the readings from the horizontal sensor, utilizing the sonic reflector, and the vertical sensor without the sonic reflector. The overall mean height reading for the vertical sensor was 365 mm, while the horizontal sensor had an overall mean height reading of 356 mm. The average calibration height (sensor setup on a concrete surface) was 417 mm. Thus, actual sensor readings were 52 mm less than the setup height for the vertical sensor, and 61 mm less than the setup height for the horizontal sensor. These deviations in sensor height readings were caused by soil compaction. Compaction of the soil under the tractor tire caused the sensor-to-target distance to be less than the setup sensor-to-target distance when travelling

Table 7.1 Analysis of variance from the cutter bar height sensor evaluation tests

Source of variance	Degree of freedom	Sum of squares	Mean square	F	
Reps	2	.039	0.019	.08	
Sensors(A)	1	2.749	2.749	12.22	**
Terrain(B)	1	51.350	51.350	228.22	**
Heights(C)	2	10978.986	5489.493	24,397.00	**
Speed (D)	2	0.141	0.071	.32	
A X B	1	1.998	1.998	8.88	**
A X C	2	2.078	1.039	4.62	*
A X D	2	0.018	0.009	0.04	
ERROR	94	21.104	0.225		
Total	107	11058.463			

** sig. at p = 1%

* sig. at p = 2.5%

over stubble and hay land terrain.

The difference in height readings between the vertical and horizontal sensors were due to the sonic reflector attachment. The horizontal sensor, utilizing the sonic reflector, caused dispersion of the sonic beam pattern, which in turn increased the target area being monitored for echos. This made the sensor more susceptible to reading crop material as the target. Thus, the vertical sensor, being less sensitive to echos returned from plant material, received echos that were produced from the ground terrain. Therefore, the vertical sensor produced a more accurate target-to-ground distance reading than the horizontal sensor. The two sensors were further studied in an anechoic chamber, where polar plots of relative intensity versus sensor direction indicated that the horizontal sensor dispersed sound pulses over a larger area than the vertical sensor.

Appendix B, which lists the number of error conditions (target echo not returned) that occurred during testing, shows that the vertical sensor had 3.6 % of its readings in error,

while the horizontal sensor had 1.4 % error readings. Although the vertical sensor gives a more accurate sensor-to-ground distance, it is more likely to fail in target detection. Since accuracy was a primary concern, the vertical sensor produced the most favorable results.

Second, the ANOVA shows a highly significant difference between terrain types. The sensors displayed a sensor-to-ground height reading that was closer to 417 mm (average calibration height) when travelling over stubble terrain than travelling over hay land terrain. The mean sensor reading over stubble terrain was 378 mm, while the mean sensor reading over hay land terrain was 343 mm. The hay land terrain, which had a denser crop cover than the stubble terrain, tended to cause a shorter sensor-to-ground distance to be read as the sensors detected the hay crop canopy as the target instead of the ground terrain. This result was in agreement with previous observations, during the field test, on the effects of green plant material.

Third, a significant difference between sensor heights was indicated. In one sense this result is meaningless, as the three different heights were fixed variables in the experimental design. However, according to Appendix B, the ability of the sensor to measure true sensor-to-ground height increased as sensor-to-ground distance increased. The coefficient of variation (CV) for each treatment decreased as sensor-to-ground height increased. Average CV's for the 130, 380, and 740 mm sensor-to-ground heights were 73.8 %, 8.0 % and 4.9 % respectively, indicating the increased accuracy of the readings with increased heights.

This trend can be explained by the fact that echos produced from targets other than the ground tended to be directed away from the sensor due to crop material orientation not being perpendicular to the sound wave vector. As sensor-to-ground height increased, the probability that an echo produced from a target other than the ground surface would be detected decreased.

Fourth, sensor-to-ground height readings were unaffected by forward travel speed. Two additional independent runs were carried out for further analysis of speed effects. Two runs with a tractor travel speed of 25 km/h, one on a highway surface, and one on a gravel road surface, indicated no effect on sensor height reading accuracy.

Test runs over summerfallow terrain were used to analyze the sensor's ability to detect a step change in elevation. Figure 7.5 shows a graph of the sensor readings vs position. Both sensor configurations were successful in detecting the step change. The step change was recognized within a 0.5 s interval, as this was the time period that would elapse due to the software polling sequence.

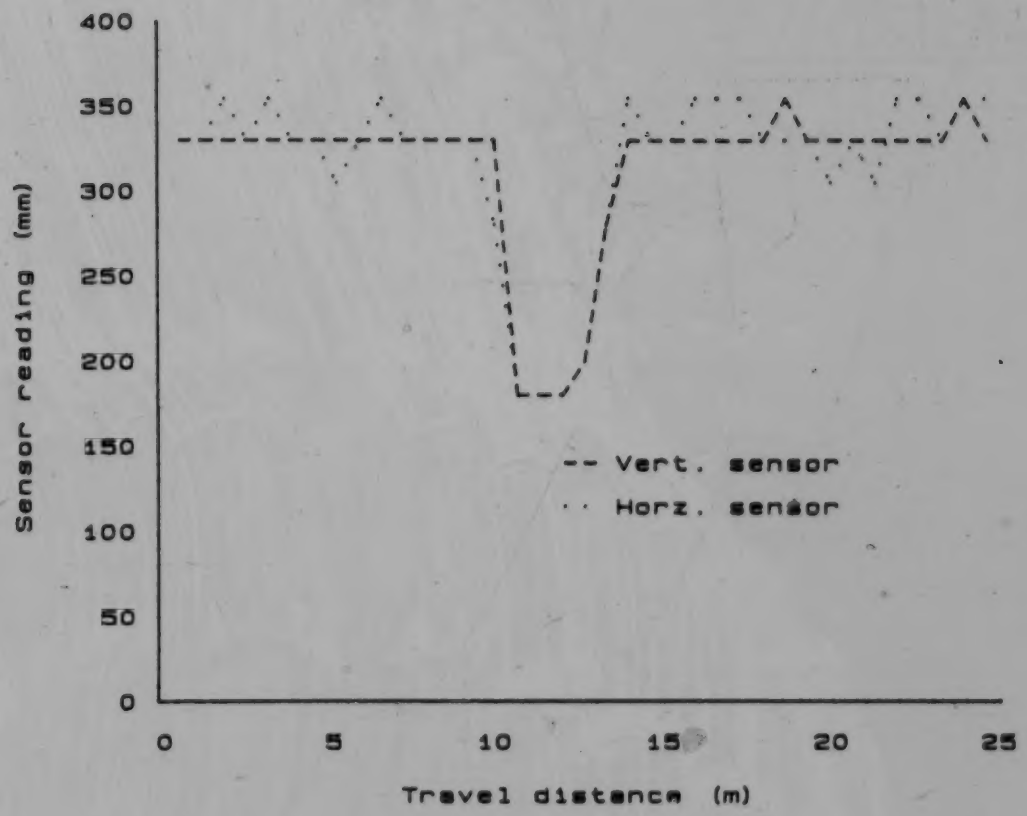


Figure 7.5 Cutter bar height sensor response to a step change in elevation

8. CONCLUSIONS

A working prototype was successfully completed and field testing has proven that the principles of sensor and control operation are correct and functional. The following conclusions were drawn from the evaluation and testing of the control device.

1. Controller hardware design fulfilled prototype and reliability demands required for field testing.
2. Controller operating system software reliably executed data acquisition, control and operator console update, and data storage tasks.
3. The cutter bar height sensor proved successful in measuring the cutter bar-to-ground height. Of the two sensor orientations used, the vertical sensor without the sonic reflector yielded a more accurate reading of the sensor-to-target distance than the horizontal sensor orientation which utilized the sonic reflector. However, the vertically orientated sensor failed to detect a target more often than the horizontally positioned sensor. Sensor accuracy was independent of implement speed, but was dependent on the terrain type and sensor-to-target distance. Accurate results were achieved over wheat stubble terrain, but sensor-to-target distance readings were low over densely vegetated hay land terrain. Sensor accuracy increased as sensor-to-target distance was increased.
4. Cutter bar height control incurred instability due to crop conditions and transient motion of the windrower. A heavy infestation of Russian thistle inhibited proper sensor height readings which diminished control effectiveness. Also, cutter bar vertical raise and lower speeds along with torsional spring action of the windrower platform caused control instability.
5. Reel height control proved effective in maintaining the reel position at a proper height with respect to the crop canopy top. Control performance deteriorated with thin crop stands.

9. RECOMMENDATIONS FOR FURTHER WORK

Design problems with the existing controller hardware have been analyzed as being minor and easily rectifiable. Power supply and shielding aspects of the microprocessor require refinement so that electrical interference produced from windrowing equipment does not affect microprocessor performance.

Current amplification circuitry within the interface junction box, used to power the solenoid-hydraulic valves, requires voltage spike protection circuitry as damage was incurred to this equipment during testing.

Operating system software performed the required tasks adequately. However, the system can be modified to run more efficiently. Improvements in the control algorithm are required to enhance control performance. The use of variable flow electro-hydraulic valves would enable proportional control strategies to be utilized in the control algorithm, which would enhance control stability.

The number of sensor readings sampled, as well as the averaging technique used, are parameters requiring further study. Averaging techniques such as "sliding window" and unequally weighted running averages should be considered.

Refinements in the cutter bar height sensor are required to improve sensor performance when operating in weedy and hay land terrain conditions. Sensor operating frequency, sound intensity, and echo amplification parameters should be investigated in detail to increase sensor reliability over a wide range of field conditions.

With regard to the crop height sensor, the number of impacts received by a piezo strip on the sensor is dependent on its area of contact with the crop. The effective frontal area of each piezo sensor is presently 500 mm^2. To increase the sensitivity of the crop sensor when windrowing very thin crops, the impact area of each piezo strip could be increased. The piezo area in contact with the crop is only limited by sensor-crop interference problems that may occur as the sensor must move through the crop.

10. REFERENCES

ASAE. 1983. D230.3 Agricultural machinery management data. Agricultural Engineers Yearbook. Am. Soc. Agric. Eng. St. Joseph, MI. pp 200-207

Coad, C.A., J.H. Ruff and C.G. Coble. 1979. Microprocessor-based ultrasonic height controller for sugarcane harvesters. ASAE Paper No. 79-1571. Am. Soc. Agric. Eng. St. Joseph, MI.

Drayfuss, H. 1960. The measure of man; human factors in design. Witney Library of Design. New York, NY.

Gunderson, D.G., T.G. Kirk, J.N. Wilson and F.B. Dyck. 1981. A comparison of ultrasonic, ski, and ski-wheel systems for tillage depth measurement. ASAE Paper No. 81-1602. Am. Soc. Agric. Eng. St. Joseph, MI.

Hoechst. 1984. Conservation For Profit. Hoechst Canada Inc., Regina, SK.

Innovator. 1987. Model J-3:16 Swather: Product Bulletin. Innovator Enterprises Inc., Box 45, D'arcy, SK.

Kaminski, T.L. and G.C. Zoerb. 1965. Automatic header-height control for grain crops. Transactions of the ASAE 8(2):284-287

Kepner, R.A., R. Bainer and E.L. Barger. 1978. Principles of farm machinery. AVI Publishing Company Inc., Westpoint, CN. pp 333, 400

Kirk, T.G. and A.E. Krause. 1975. Swather edge guide steering control system. ASAE Paper No. 75-1079. Am. Soc. Agric. Eng. St. Joseph, MI.

Macdon. 1986. Product Bulletin. Macdon Industries Ltd, Winnipeg, MB.

Motorola. 1981. Motorola Microprocessors Data Manual. Motorola Semiconductor Products Inc., Austin, TX.

O'Sullivan, J.A. 1985. Review of non-contact proximity transducers for use in a sprayer boom height control system. Divisional Note 1256. National Institute of Agricultural Engineering. Silsoe, Bedford.

Pask, G.S., J.N. Wilson and G.C. Zoerb. 1973. An automatic height-of-cut control system for windrowers. ASAE Paper No. 73-155. Am. Soc. Agric. Eng. St. Joseph, MI.

Pennwalt. 1983. Kynar piezo film technical manual. Pennwalt Corporation, King of Prussia, PA.

PAMI. 1977a. Evaluation Report No. E0576A:International Harvester 914 Pull-Type Combine 11 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1977b. Evaluation Report No. E0576C:John Deere Sidehill 6600 Self-Propelled Combine 11 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1977c. Evaluation Report No. E1876A:Co-op Implements 550SP swather 6 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1978. Evaluation Report No. E0777A:Sperry New Holland TR70 Self-Propelled Combine 13 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1981a. Evaluation Report No. E318OB:Allis-Chalmers Gleaner N-6 Self-Propelled Combine 11 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1981b. Evaluation Report No. E318OD:John Deere 8820 Self-propelled Combine 11 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1982. Evaluation Report No. 288:White 9700 Self-Propelled Combine 13 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1983. Evaluation Report No. 312:Claas Dominator 106 Self-Propelled Combine 13 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1984. Evaluation Report No. 359:Wallis PTL 130 Double Windrow Attachment 6 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1986. Evaluation Report No. 474:New Holland TR96 Self-Propelled Combine 13 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1987a. Evaluation Report No. 531:Case IH 1660 Self-Propelled Combine 13 pp. P.O. Box 1900, Humboldt, SK.

PAMI. 1987b. Evaluation Report No. 532:Versatile Trans-Axial 2000 Pull-Type Combine 13 pp. P.O. Box 1900, Humboldt, SK.

Polaroid. 1984a. Polaroid ultrasonic ranging system handbook application notes/technical papers. Polaroid Canada Inc. Rexdale, ON.

Polaroid. 1984b. Ultrasonic ranging system. Polaroid Canada Inc. Rexdale, ON.

Senstek. 1987. Header height control:Product Bulletin. Senstek, P.O. Box 340 Saskatoon, SK.

Singh, K., L.C. Buchanan and A. De Groot. 1984. An ultrasonic depth sensor for tillage implements. ASAE Paper No. 84-101. Am. Soc. Agric. Eng. St. Joseph, MI.

Smith, H.P. 1964. Farm machinery and equipment. 5th Ed. Mcgraw Hill, New York, NY. p. 335

Thornley, W.R., H. Park, J.G. Wiser and E.C. Lundahl. 1985. Automatic sonar activated height control for a header. United States Patent No. 4507910

11. APPENDIX A. CONTROLLER OPERATING SYSTEM PROGRAM

```
                    1  *       CUTTER BAR AND REEL HEIGHT CONTROLLER SOFTWARE
                    2  *
                    3  *       PROGRAMMER:     ROY MAKI
                    4  *       DATE:           FALL 1987
                    5  *       VERSION:        1.0
                    6  *       PROCESSOR:      MC6809 (MOTOROLA)
                    7  *
                    8  **********VARIABLE IDENTIFICATION AND INITIALIZATION****
                    9  *
                   10  *       ASCII VALUES USED FOR LCD CHARACTER DISPLAY
                   11  *
0041               12  ACHAR   EQU  $41          *'A'
0042               13  BCHAR   EQU  $42          *'B'
0043               14  CCHAR   EQU  $43          *'C'
0044               15  DCHAR   EQU  $44          *'D'
0045               16  ECHAR   EQU  $45          *'E'
004C               17  LCHAR   EQU  $4C          *'L'
004D               18  MCHAR   EQU  $4D          *'M'
004E               19  NCHAR   EQU  $4E          *'N'
0052               20  RCHAR   EQU  $52          *'R'
0053               21  SCHAR   EQU  $53          *'S'
0054               22  TCHAR   EQU  $54          *'T'
0055               23  UCHAR   EQU  $55          *'U'
0020               24  BLANK   EQU  $20          *' '
                   25  *
                   26  *       BUFFERS USED TO STORE ASCII CHARACTER VALUES
                   27  *
D07A               28  SCOUNT  EQU  $D07A        *DATA SEND COUNTER
D072               29  OFFSET  EQU  $D072        *ACIA ADDRESS OFFSET MEMORY LOCATION
D070               30  REELAM  EQU  $D070        *REEL AUTO-MANUAL FLAG
D050               31  DISBUF  EQU  $D050        *DISPLAY BUFFER GOES TO DISBUF+15
D045               32  CHRBUF  EQU  $D045        *SINGLE CHARACTER BUFFER
D048               33  TEMP1   EQU  $D048        *1ST ECHO TEMPORARY STORAGE
D047               34  NOECHO  EQU  $D047        *CB ERROR FLAG
D046               35  ALLFGS  EQU  $D046        *MODFGS MODIFIED BUFFER
D044               36  PIPELH  EQU  $D044        *MULTIPLICATION CONSTANT FOR CB HEIGHT
D043               37  MODPOS  EQU  $D043        *SENSOR SET POINTER USED WITH ALLFGS
D042               38  SPLIM   EQU  $D042        *UPPER SETPOINT LIMIT REGISTER
D041               39  MODFGS  EQU  $D041        *OPERATOR SWITCH STATUS REGISTER
D040               40  PCOUNT  EQU  $D040        *NUMBER OF POLLING CYCLE FLAG
```

```
                    42  *
                    43  *       VARIABLES STORED IN RAM FOR EACH SENSOR BLOCK
                    44  *
                    45  *       D SENSOR SET
                    46  *
D03F                47  DCBSL2 EQU  $D03F       *SECOND ECHO LOW BYTE
D03E                48  DCBSH2 EQU  $D03E       *SECOND ECHO HIGH BYTE
D03A                49  DMULTL EQU  $D03A       *MULT LOW BYTE
D039                50  DMULTM EQU  $D039       *MULT MEDIUM BYTE
D038                51  DMULTH EQU  $D038       *MULT HIGH BYTE
D037                52  DCBERR EQU  $D037       *CB ERROR
D036                53  DCBSP  EQU  $D036       *CB SET POINT (BCD) RANGE 0 TO 99
D035                54  DCBACT EQU  $D035       *CB ACTUAL VALUE (BCD) RANGE 0 TO 99
D034                55  DCBSML EQU  $D034       *FIRST ECHO LOW BYTE
D033                56  DCBSMH EQU  $D033       *FIRST ECHO HIGH BYTE
D032                57  DRSP   EQU  $D032       *REEL SET POINT (BCD) RANGE 0 TO 9
D031                58  DRACT  EQU  $D031       *ACTUAL REEL VALUE (BCD) RANGE 0 TO 9
D030                59  DRSUM  EQU  $D030       *REEL SUM
                    60  *
                    61  *       C SENSOR SET
                    62  *
D02F                63  CCBSL2 EQU  $D02F       *SECOND ECHO LOW BYTE
D02E                64  CCBSH2 EQU  $D02E       *SECOND ECHO HIGH BYTE
D02A                65  CMULTL EQU  $D02A       *MULT LOW BYTE
D029                66  CMULTM EQU  $D029       *LOW MEDIUM BYTE
D028                67  CMULTH EQU  $D028       *MULT HIGH BYTE
D027                68  CCBERR EQU  $D027       *CB ERROR
D026                69  CCBSP  EQU  $D026       *CB SET POINT (BCD) RANGE 0 TO 99
D025                70  CCBACT EQU  $D025       *CB ACTUAL VALUE (BCD) RANGE 0 TO 99
D024                71  CCBSML EQU  $D024       *FIRST ECHO LOW BYTE
D023                72  CCBSMH EQU  $D023       *FIRST ECHO HIGH BYTE
D022                73  CRSP   EQU  $D022       *REEL SET POINT (BCD) RANGE 0 TO 9
D021                74  CRACT  EQU  $D021       *ACTUAL REEL VALUE (BCD) RANGE 0 TO 9
D020                75  CRSUM  EQU  $D020       *REEL SUM
```

```
                    77  *
                    78  *       B SENSOR SET
                    79  *
D01F                80  BCBSL2 EQU  $D01F       *SECOND ECHO LOW BYTE
D01E                81  BCBSH2 EQU  $D01E       *SECOND ECHO HIGH BYTE
D01A                82  BMULTL EQU  $D01A       *MULT LOW BYTE
D019                83  BMULTM EQU  $D019       *MULT MEDIUM BYTE
D018                84  BMULTH EQU  $D018       *MULT HIGH BYTE
D017                85  BCBERR EQU  $D017       *CB ERROR
D016                86  BCBSP  EQU  $D016       *CB SET POINT (BCD) RANGE 0 TO 99
D015                87  BCBACT EQU  $D015       *CB ACTUAL VALUE (BCD) RANGE 0 TO 99
D014                88  BCBSML EQU  $D014       *FIRST ECHO LOW BYTE
D013                89  BCBSMH EQU  $D013       *FIRST ECHO HIGH BYTE
D012                90  BRSP   EQU  $D012       *REEL SET POINT (BCD) RANGE 0 TO 9
D011                91  BRACT  EQU  $D011       *ACTUAL REEL VALUE (BCD) RANGE 0 TO 9
D010                92  BRSUM  EQU  $D010       *REEL SUM
                    93  *
                    94  *       A SENSOR SET
                    95  *
D00F                96  ACBSL2 EQU  $D00F       *SECOND ECHO LOW BYTE
D00E                97  ACBSG2 EQU  $D00E       *SECOND ECHO HIGH BYTE
D00A                98  AMULTL EQU  $D00A       *MULT LOW BYTE
D009                99  AMULTM EQU  $D009       *MULT MEDIUM BYTE
D008               100  AMULTH EQU  $D008       *MULT HIGH BYTE
D007               101  ACBERR EQU  $D007       *CB ERROR
D006               102  ACBSP  EQU  $D006       *CB SET POINT (BCD) RANGE 0 TO 99
D005               103  ACBACT EQU  $D005       *CB ACTUAL VALUE (BCD) RANGE 0 TO 99
D004               104  ACBSML EQU  $D004       *FIRST ECHO LOW BYTE
D003               105  ACBSMH EQU  $D003       *FIRST ECHO HIGH BYTE
D002               106  ARSP   EQU  $D002       *REEL SET POINT (BCD) RANGE 0 TO 9
D001               107  ARACT  EQU  $D001       *ACTUAL REEL VALUE (BCD) RANGE 0 TO 9
D000               108  ARSUM  EQU  $D000       *REEL SUM
                   109  *
                   110  *       PIA REGISTER VARIABLES AND ADDRESSES
                   111  *
C000               112  PIAADR EQU  $C000       *A SENSOR SET I/O DATA REGISTER
C001               113  PIAACR EQU  PIAADR+1    *A SENSOR SET CONTROL REGISTER
C010               114  PIABDR EQU  $C010       *B SENSOR SET I/O DATA REGISTER
C011               115  PIABCR EQU  PIABDR+1    *B SENSOR SET CONTROL REGISTER
C020               116  PIACDR EQU  $C020       *C SENSOR SET I/O DATA REGISTER
C021               117  PIACCR EQU  PIACDR+1    *C SENSOR SET CONTROL REGISTER
C030               118  PIADDR EQU  $C030       *D SENSOR SET I/O DATA REGISTER
C031               119  PIADCR EQU  PIADDR+1    *D SENSOR SET CONTROL REGISTER
8000               120  PIASWD EQU  $8000       *OPERATOR SWITCH INPUT PIA
8001               121  PIASWC EQU  PIASWD+1    *OPERATOR SWITCH INPUT CONTROL REGISTER
8002               122  ATPIAD EQU  PIASWD+2    *OPERATOR AUTO/MANUAL SWITCH PIA
8003               123  ATPIAC EQU  PIASWD+3    *OPERATOR AUTO/MANUAL SWITCH CONTROL REG.
```

```
                        125  *
                        126  *       PTM REGISTER  VARIABLES AND ADDRESSES
                        127  *
E000                    128  PTMCR1 EQU  $E000       *PTM TIMER #1 CONTROL REGISTER ADDRESS
E001                    129  PTMCR2 EQU  PTMCR1+1    *PTM TIMER #2 CONTROL REGISTER ADDRESS
E000                    130  PTMCR3 EQU  PTMCR1      *PTM TIMER #3 CONTROL REGISTER ADDRESS
E001                    131  PTMSR  EQU  PTMCR1+1    *PTM INTERRUPT STATUS REGISTER ADDRESS
E002                    132  LATCH1 EQU  PTMCR1+2    *PTM TIMER #1 LATCH REGISTER ADDRESS
E004                    133  LATCH2 EQU  PTMCR1+4    *PTM TIMER #2 LATCH REGISTER ADDRESS
E006                    134  LATCH3 EQU  PTMCR1+6    *PTM TIMER #3 LATCH REGISTER ADDRESS
E002                    135  COUNT1 EQU  LATCH1      *PTM TIMER #1 COUNT REGISTER ADDRESS
E004                    136  COUNT2 EQU  LATCH2      *PTM TIMER #2 COUNT REGISTER ADDRESS
E006                    137  COUNT3 EQU  LATCH3      *PTM TIMER #3 COUNT REGISTER ADDRESS
                        138  *
                        139  *       ACIA REGISTER VARIABLES AND ADDRESSES
                        140  *
E008                    141  ACIACR EQU $E008        *ACIA CONTROL REGISTER ADDRESS
E008                    142  ACIASR EQU $E008        *ACIA STATUS REGISTER ADDRESS
E009                    143  ACIADR EQU $E009        *ACIA TRANSMIT DATA REGISTER ADDRESS
                        144  *
                        145  *       LCD VARIABLES
                        146  *
0089                    147  INCCSR EQU  %10001001   *LCD INCREMENT CURSOR FLAG
0084                    148  GETCHR EQU  %10000100   *LCD READ CHARACTER FLAG
0000                    149  LCD    EQU  $0000       *LIQUID CRYSTAL DISPLAY ADDRESS
                        150  *
                        151  *       STACK POINTERS INITIAL LOCATIONS
                        152  *
D600                    153  SSTACK EQU  $D600       *INITIALIZE THE SYSTEM STACK
D7FF                    154  USTACK EQU  $D7FF       *INITIALIZE THE USER STACK
                        155  *
                        156  *       INTERRUPT AND RESET VECTOR ADDRESSES
                        157  *
FFFE                    158           ORG  $FFFE
FFFE F80A               159           FDB  RESET
FFF6                    160           ORG  $FFF6      *INITIALIZE FIRQ INTERRUPT VECTOR
FFF6 F92B               161           FDB  FIRQ
FFF8                    162           ORG  $FFF8      *INITIALIZE IRQ INTERRUPT VECTOR
FFF8 F969               163           FDB  IRQ
                        164  *
                        165  *       LCD DISPLAY INITIALIZATION PARAMETERS
                        166  *
F800                    167           ORG  $F800
F800 686B616D6264       168  FLAGS    FCB  $68,$6B,$61,$6D,$62,$64,$67,$6E,$00,$8A
F806 676E008A
```

```
                          170  ************************************************************
                          171  *  << MAIN PROGRAM LOOP >>                                 *
                          172  *                                                          *
                          173  *    THE MAIN ROUTINE SYNCHRONIZES AND CONTROLS THE SOFTWARE  *
                          174  *    OF THE PROCESSOR OPERATION                            *
                          175  ************************************************************
                          176  *
                          177  *    LOAD THE USER AND HARDWARE STACK REGISTERS
                          178  *
F80A CE    D7FF      3    179  RESET   LDU #USTACK
F80D 10CE D600       4    180         LDS #SSTACK
                          181  *
                          182  *    CLEAR MEMORY LOCATIONS IN RAM
                          183  *
F811 8E    D000      3    184         LDX #$D000
F814 6F    80        8    185  RAMCLR  CLR ,X+              *CLEAR EACH RAM LOCATION
F816 8C    D7FF      4    186         CMPX #$D7FF
F819 26    F9        3    187         BNE RAMCLR
                          188  *
                          189  *    JSR INIT TO INITIALIZE PIA, PTM,LCD, AND LOAD SETPOINTS
                          190  *
F81B BD    F842      8    191         JSR INIT
                          192  *
                          193  *     BEGIN THE MAIN LOOP
                          194  *
F81E 86    08        2    195  BEGIN   LDA #$08
F820 B7    D040      5    196         STA PCOUNT           *INITIALIZE THE POLL COUNTER
F823 7A    D040      7    197  MLOOP   DEC PCOUNT           *DEC PCOUNT EVERY ROUND OF MLOOP
F826 BD    F8F8      8    198         JSR POLCR            *POLL ALL REEL AND CB SENSORS ONCE
F829 7D    D040      7    199         TST PCOUNT           *8 POLLING SEQUENCES COMPLETED?
F82C 26    F5        3    200         BNE MLOOP            *NO POLL AGAIN
F82E BD    F9B4      8    201         JSR AUTMAN           *CHECK FOR AUTO OR MANUAL CONTROL
F831 BD    F9D7      8    202         JSR ABCD             *CHECK IF A, B, C, D SWITCHES ARE PUSHED
F834 BD    FA3F      8    203         JSR ADJUST           *SET MODFGS ACCORDING TO SWITCH STATUS
F837 BD    FA7E      8    204         JSR DISPLY           *SET UP DISPLAY BUFFER
F83A BD    FC0C      8    205         JSR RFRSH            *REFRESH THE LCD SCREEN
F83D BD    FC59      8    206         JSR COMPAR           *INITIATE CONTROL ACTION
F840 20    DC        3    207         BRA BEGIN            *RESTART DATA ACQUISITION SEQUENCE
```

```
                        209  *  << SUBROUTINE INIT >>
                        210  *
                        211  *    THIS SUBROUTINE INITIALIZES THE PIA, PTM, LCD, AND
                        212  *    INITIALIZES CB AND REEL SETPOINTS
                        213  *
                        214  *
                        215  *       INITIALIZE PIA'S FOR DATA I/O
                        216  *
F842 4F                2  217  INIT    CLRA
F843 B7   C001         5  218          STA PIAACR          *SELECT DATA DIRECTION REGISTER
F846 B7   C011         5  219          STA PIABCR
F849 B7   C021         5  220          STA PIACCR
F84C B7   C031         5  221          STA PIADCR
F84F B7   8001         5  222          STA PIASWC
F852 B7   8003         5  223          STA ATPIAC
F855 86   F0           2  224          LDA #%11110000      *BITS 0,1,2,3 ARE SET FOR INPUT
F857 B7   C000         5  225          STA PIAADR          *BITS 4,5,6,7 ARE SET FOR OUTPUT
F85A B7   C010         5  226          STA PIABDR          *FOR SENSOR I/O PIA'S
F85D B7   C020         5  227          STA PIACDR
F860 B7   C030         5  228          STA PIADDR
F863 4F                2  229          CLRA                *SWITCH INPUT PIA SET FOR INPUTS
F864 B7   8000         5  230          STA PIASWD          *WITH PB7 SET AS AN OUTPUT TO
F867 86   C8           2  231          LDA #%11001000      *ENABLE THE SOLENOID CONTROL VALVE
F869 B7   8002         5  232          STA ATPIAD          *OUTPUT LINES, PB6 FOR CONSOLE ENABLE
F86C 86   34           2  233          LDA #%00110100      *ACCESS DATA REGISTER AND SET CA2 LOW
F86E B7   C001         5  234          STA PIAACR          *AND SET CONTROL REGISTERS
F871 B7   C011         5  235          STA PIABCR          *FOR SENSOR I/O PIA'S
F874 B7   C021         5  236          STA PIACCR
F877 B7   C031         5  237          STA PIADCR
F87A 86   04           2  238          LDA #%00000100      *SET SWITCH INPUT PIA TO ACCESS
F87C B7   8001         5  239          STA PIASWC          *DATA REGISTER
F87F B7   8003         5  240          STA ATPIAC
F882 4F                2  241          CLRA                *INITIALLY CLEAR DATA REGISTERS
F883 B7   C000         5  242          STA PIAADR          *FOR SENSOR I/O PIA'S
F886 B7   C010         5  243          STA PIABDR
F889 B7   C020         5  244          STA PIACDR
F88C B7   C030         5  245          STA PIADDR
F88F 86   48           2  246          LDA #%01001000      *DISABLE CONSOLE DATA BUS AND
F891 B7   8002         5  247          STA ATPIAD          *ENABLE SOLENOID OUTPUTS
```

```
                          249  *            PTM INITIALIZATION BLOCK
                          250  *
F894 8E   3004     3      251       LDX #12500          *LOAD LATCH 1 WITH 12500 (14ms)
F897 BF   E002     6      252       STX LATCH1
F89A 8E   32C8     3      253       LDX #13000          *LOAD LATCH 2 WITH 13000
F89D BF   E004     6      254       STX LATCH2
F8A0 BF   E006     6      255       STX LATCH3          *LOAD LATCH 3 WITH 13000
                          256  *
                          257  *    TIMER #3 USED FOR TIMING THE SECOND ECHO
                          258  *
F8A3 86   4A       2      259       LDA #%01001010      *TIMER #3: DIVIDE BY ONE
F8A5 B7   E000     5      260       STA PTMCR3          *FREQ. COMPARISON MODE, 16 BIT COUNT,
                          261  *                        *INTERRUPTS ENABLED
                          262  *                        *TIMER OUTPUT LINE DISABLED
                          263  *                        *INTERRUPT OCCURS IF TIME OUT OCCURS
                          264  *                        *BEFORE TIMEOUT COUNT
                          265  *
                          266  *    TIMER #2 USED FOR TIMING THE FIRST ECHO
                          267  *
F8A8 86   4B       2      268       LDA #%01001011      *TIMER #2: CR1 ACCESS ENABLED
F8AA B7   E001     5      269       STA PTMCR2          *FREQ. COMPARISON MODE, 16 BIT COUNT,
                          270  *                        *INTERRUPTS ENABLED
                          271  *                        *TIMER OUTPUT LINE DISABLED
                          272  *                        *INTERRUPT OCCURS IF GATE GOES LOW
                          273  *                        *BEFORE TIMEOUT OCCURS
                          274  *
                          275  *    TIMER #1 USED FOR TIME OUT ERROR DETECTION
                          276  *
F8AD 86   6A       2      277       LDA #%01101010      *TIMER #1: TIMER SET TO BEGIN
F8AF B7   E000     5      278       STA PTMCR1          *INTERRUPTS ENABLED
                          279  *                        *TIMER OUTPUT LINE DISABLED
                          280  *                        *INTERRUPT OCCURS IF TIME OUT OCCURS
                          281  *                        *BEFORE GATE INPUT GOES LOW
```

```
                                283  *
                                284  *        ACIA INITIALIZATION BLOCK
                                285  *
F8B2 86   43       2    286          LDA #%01000011     *MASTER RESET ACIA
F8B4 B7   E008     5    287          STA ACIACR
F8B7 86   41       2    288          LDA #%01000001     *ACIA SET FOR DIVIDE BY 16,
F8B9 B7   E008     5    289          STA ACIACR         *7 BITS, EVEN PARITY, 2 STOP BITS
                                290  *
                                291  *        LCD DISPLAY INITIALIZATION BLOCK
                                292  *
F8BC 7F   8002     7    293          CLR ATPIAD         *ENABLE THE CONSOLE DATA BUS
F8BF 8E   F800     3    294          LDX #FLAGS         *LOAD ALL INITIALIZATION FLAGS
F8C2 A6   80       6    295  LDFLG   LDA ,X+            *POINTED AT BY THE INDEX REGISTER
F8C4 97   00       4    296          STA LCD
F8C6 81   8A       2    297          CMPA #$8A          *END WHEN THE LAST FLAG IS LOADED
F8C8 26   F8       3    298          BNE LDFLG
F8CA 0F   00       6    299          CLR LCD
F8CC 86   48       2    300          LDA #%01001000     *DISABLE THE CONSOLE DATA BUS
F8CE B7   8002     5    301          STA ATPIAD         *AND SET THE FLIPFLOP CLEAR HIGH
                                302  *
                                303  *        LOAD STARTUP CB AND REEL SETPOINTS
                                304  *
F8D1 86   10       2    305          LDA #$10           *SET CB SETPOINTS AT 250 MM HEIGHT
F8D3 B7   D006     5    306          STA ACBSP
F8D6 B7   D016     5    307          STA BCBSP
F8D9 B7   D026     5    308          STA CCBSP
F8DC B7   D036     5    309          STA DCBSP
F8DF 86   01       2    310          LDA #$01           *SET REEL SETPOINTS AT 1
F8E1 B7   D002     5    311          STA ARSP
F8E4 B7   D012     5    312          STA BRSP
F8E7 B7   D022     5    313          STA CRSP
F8EA B7   D032     5    314          STA DRSP
F8ED 86   74       2    315          LDA #116           *LOAD THE MULTIPLIER
F8EF B7   D044     5    316          STA PIPELH
F8F2 86   07       2    317          LDA #07            *OFFSET EQU 7
F8F4 B7   D072     5    318          STA OFFSET
F8F7 39            5    319          RTS                *RETURN TO THE MAIN ROUTINE
```

```
                        321  *  << SUBROUTINE POLCR >>
                        322  *
                        323  *      THIS ROUTINE ACTIVATES THE CB SENSOR, READS AND STORES
                        324  *      THE REEL HEIGHT SENSOR, AND WAITS FOR AN INTERRUPT TO
                        325  *      OCCUR.  ALL CB AND REEL SENSORS ARE POLLED ONCE.
                        326  *
F8F8 36   10       5    327  POLCR   PSHU X              *STORE CURRENT VALUE ON USTACK
F8FA 8E   C000     3    328          LDX #$C000         *LOAD X REGISTER WITH PIA BASE ADDRESS
F8FD 86   3C       2    329  NXTSET  LDA #%00111100     *SET C*2 HIGH TO ENABLE CB SENSOR
F8FF A7   01       5    330          STA $1,X
F901 A6   84       4    331          LDA ,X             *READ REEL SENSOR
F903 84   0F       2    332          ANDA #%00001111
F905 AB   89 1000  8    333          ADDA $1000,X       *ADD READING TO THE REEL SUM REGISTER
F909 A7   89 1000  8    334          STA  $1000,X       *AND STORE IT IN SUM REGISTER
F90D BD   FF86     8    335          JSR SEND           *SEND DATA TO CASSETTE TAPE
F910 3C   BF      21    336          CWAI #$BF          *UNMASK FIRQ AND WAIT FOR 1ST ECHO
F912 1A   40       3    337          ORCC #%01000000    *MASK THE FIRQ BIT
F914 7D   D047     7    338          TST NOECHO         *FIRST ECHO UNDETECTED?
F917 26   04       3    339          BNE RNDIRQ         *THEN SKIP OVER THE SECOND INTERRUPT
F919 3C   EF      21    340          CWAI #$EF          *UNMASK IRQ AND WAIT FOR 2ND ECHO
F91B 1A   10       3    341          ORCC #%00010000    *MASK THE IRQ BIT
F91D 7F   D047     7    342  RNDIRQ  CLR NOECHO         *RESET THE NOECHO FLAG FOR NEXT SENSOR
F920 30   88 10   5    343          LEAX $10,X         *INCREMENT X TO READ THE NEXT SENSOR SET
F923 8C   C040     4    344          CMPX #$C040        *ALL 4 SENSOR SETS READ?
F926 26   D5       3    345          BNE NXTSET         *NO, READ THE NEXT SENSOR SET
F928 37   10       5    346          PULU X             *YES: RESTORE PREVIOUS X AND RETURN TO
F92A 39            5    347          RTS                *THE MAIN ROUTINE
```

```
                                  349  *  << FAST INTERRUPT SERVICING ROUTINE >>
                                  350  *
                                  351  *      THIS ROUTINE DETERMINES WHICH PTM TIMER CAUSED THE
                                  352  *      INTERRUPT. IF TIMER #2 CAUSED THE INTERRUPT, TIMER #2
                                  353  *      COUNT IS READ AND STORED IN THE *CBSUM REGISTER
                                  354  *      IF TIMER #1 CAUSED THE INTERRUPT, *CBERR REGISTER
                                  355  *      IS DECREMENTED, THE NOECHO FLAG IS SET, AND ALL TIMERS
                                  356  *      ARE RESET FOR THE NEXT SENSOR READING
                                  357  *
F92B B6   E001      5   358  FIRQ    LDA PTMSR          *CHECK STATUS TO DETERMINE WHICH TIMER
F92E 44             2   359          LSRA               *CAUSED THE INTERRUPT
F92F 25   08        3   360          BCS ERROR          *BRANCH TO ERROR IF TIMER #1 INTERRUPTED
F931 CC   32C8      3   361          LDD #13000         *TIMER #2 CAUSED THE INTERRUPT
F934 B3   E004      7   362          SUBD COUNT2        *READ COUNT FROM THE 1ST ECHO
F937 FD   D048      6   363          STD TEMP1          *AND STORE IT IN TEMPORARY MEMORY
F93A 20   2C        3   364          BRA QUIT           *THEN END
F93C 86   34        2   365  ERROR   LDA #%00110100     *SET C*2 LOW FOR THE NEXT SENSOR ENABLE
F93E A7   01        5   366          STA $1,X
F940 BD   FC4D      8   367          JSR SDELAY         *THEN WAIT FOR FIRST ECHO TO RETURN LOW
F943 FC   E002      6   368          LDD COUNT1         *SINCE THE 1ST ECHO WAS UNDETECTED,
F946 FC   E004      6   369          LDD COUNT2         *AN ERROR EXISTS
F949 FC   E006      6   370          LDD COUNT3         *READ EACH TIMER COUNT AND
F94C CC   3004      3   371          LDD #12500         *RELOAD THE LATCHES
F94F FD   E002      6   372          STD LATCH1         *TO RESET THE TIMERS FOR THE
F952 CC   32C8      3   373          LDD #13000         *NEXT SENSOR READ SEQUENCE
F955 FD   E004      6   374          STD LATCH2
F958 FD   E006      6   375          STD LATCH3
F95B 7C   D047      7   376          INC NOECHO         *FLAG THE ERROR CONDITION
F95E 6A   89 1007  10   377          DEC $1007,X        *AND RECORD THE ERROR IN *CBERR
F962 7F   D048      7   378          CLR TEMP1
F965 7F   D049      7   379          CLR TEMP1+1
F968 3B            15   380  QUIT    RTI                *RETURN FROM INTERRUPT BACK TO POLCR
```

```
                                382  *  << REGULAR INTERRUPT SERVICING ROUTINE >>
                                383  *
                                384  *      THIS ROUTINE DETERMINES WHICH PTM TIMER CAUSED THE
                                385  *      INTERRUPT.  IF TIMER #3 CAUSED THE INTERRUPT, THE
                                386  *      SECOND ECHO INFO. IS READ FROM COUNT3 AND STORED IN
                                387  *      *CBSUM2 REGISTERS, THE 1ST ECHO INFO. IS PULLED FROM
                                388  *      THE STACK AND STORED IN *CBSUM1 REGISTERS.
                                389  *      IF TIMER #1 CAUSED THE INTERRUPT WHICH SIGNALS AN
                                390  *      ERROR CONDITION, THE *CBERR REGISTER IS DECREMENTED.
                                391  *      ALL TIMERS ARE THEN RESET FOR THE NEXT SENSOR SEQUENCE.
                                392  *
F969 B6   E001      5   393  IRQ      LDA  PTMSR        *CHECK STATUS TO DETERMINE WHICH TIMER
F96C 44                 2   394       LSRA              *CAUSED THE INTERRUPT
F96D 25   1B            3   395       BCS  ERROR2       *BRANCH TO ERROR2 IF TIMER #1 INTERRUPTED
F96F CC   32C8          3   396       LDD #13000        *CALCULATE TIMER #3 TIME(2ND ECHO)
F972 B3   E006          7   397       SUBD COUNT3       *AND ADD IT IN *CBSUM2 REGISTERS
F975 E3   89 100E      10   398       ADDD $100E,X
F979 ED   89 100E       9   399       STD  $100E,X
F97D FC   D048          6   400       LDD  TEMP1        *GET THE 1ST ECHO TIME FROM RAM
F980 E3   89 1003      10   401       ADDD $1003,X      *AND ADD IT TO THE *CBSUM1 REGISTERS
F984 ED   89 1003       9   402       STD  $1003,X
F988 20   04           3   403       BRA  OVER1
F98A 6A   89 1007      10   404  ERROR2   DEC  $1007,X      *IF AN ERROR OCCURRED, UPDATE *CBERR
F98E 86   34            2   405  OVER1    LDA #%00110100    *SET C*2 LOW FOR THE NEXT SENSOR ENABLE
F990 A7   01            5   406       STA $1,X
F992 BD   FC4D          8   407       JSR SDELAY        *THEN WAIT FOR SECOND ECHO TO RETURN LOW
F995 FC   E002          6   408       LDD  COUNT1       *FINALLY READ AND WRITE THE 3 COUNTERS
F998 FC   E004          6   409       LDD  COUNT2       *AND LATCHES TO RESET THE TIMERS FOR
F99B FC   E006          6   410       LDD  COUNT3       *THE NEXT SENSOR READ SEQUENCE.
F99E CC   3004          3   411       LDD  #12500
F9A1 FD   E002          6   412       STD  LATCH1
F9A4 CC   32C8          3   413       LDD  #13000
F9A7 FD   E004          6   414       STD  LATCH2
F9AA FD   E006          6   415       STD  LATCH3
F9AD 7F   D048          7   416       CLR TEMP1
F9B0 7F   D049          7   417       CLR TEMP1+1
F9B3 3B                15   418       RTI
```

```
                                420  *  <<  SUBROUTINE AUTMAN>>
                                421  *
                                422  *      SETS MODFGS ACCORDING TO THE POSITION OF THE
                                423  *      AUTO MANUAL SWITCH
                                424  *
F9B4 F6   D041        5         425  AUTMAN LDB  MODFGS      *GET THE FLAGS DATA
F9B7 B6   8002        5         426         LDA  ATPIAD     *GET THE SWITCH STATUS
F9BA 43               2         427         COMA
F9BB 47               2         428         ASRA
F9BC 25   07          3         429         BCS  AUTO1      *CB SWITCH IN AUTO?
F9BE C4   7F          2         430         ANDB #%01111111 *NO, RESET AUTOFLAG
F9C0 F7   D041        5         431         STB  MODFGS
F9C3 20   05          3         432         BRA  REELCK     *THEN CHECK THE REEL STATUS
F9C5 CA   80          2         433  AUTO1  ORB  #%10000000 *YES, SET CB AUTOFLAG
F9C7 F7   D041        5         434         STB  MODFGS     *SAVE THE UPDATES STATUS
F9CA 47               2         435  REELCK ASRA            *NOW CHECK THE REEL SWITCH
F9CB 25   04          3         436         BCS AUTO2       *IF REEL IS IN AUTO THEN SET REELAM FLAG
F9CD 7F   D070        7         437         CLR REELAM      *ELSE, CLEAR THE REEL AM FLAG FOR MANUAL
F9D0 39               5         438         RTS             *THEN END
F9D1 86   01          2         439  AUTO2  LDA #01         *SET THE REELAM FLAG FOR AUTO
F9D3 B7   D070        5         440         STA REELAM
F9D6 39               5         441         RTS             *THEN END
```

```
                          443  *  <<  SUBROUTINE ABCD>>
                          444  *
                          445  *      CHECKS FOR A SWITCH A THROUGH D AND SETS OR RESETS
                          446  *      CORRESPONDING FLAG.  THE FIRST SWITCH PRESSED WILL
                          447  *      CONTINUE TO BE SERVICED UNTIL IT IS RELEASED
                          448  *
F9D7 B6   8000      5     449  ABCD   LDA  PIASWD      *GET THE SWITCH STATUS
F9DA 43             2     450         COMA
F9DB F6   D041      5     451         LDB  MODFGS      *GET THE FLAGS STATUS
F9DE 57             2     452         ASRB
F9DF 25   09        3     453         BCS  SWA         *IF A FLAG GOTO SWA
F9E1 57             2     454         ASRB
F9E2 25   1B        3     455         BCS  SWB         *IF B FLAG GOTO SWB
F9E4 57             2     456         ASRB
F9E5 25   2D        3     457         BCS  SWC         *IF C FLAG GOTO SWC
F9E7 57             2     458         ASRB
F9E8 25   3F        3     459         BCS  SWD         *IF D FLAG GOTO SWD
F9EA 85   01        2     460  SWA    BITA #%00000001
F9EC 27   09        3     461         BEQ  NOTSWA      *SWITCH A SET?
F9EE F6   D041      5     462         LDB  MODFGS      *YES, SET A FLAG
F9F1 CA   01        2     463         ORB  #%00000001  *     MODFGS(0)=1
F9F3 F7   D041      5     464         STB  MODFGS
F9F6 39             5     465         RTS              *AND STOP
F9F7 F6   D041      5     466  NOTSWA LDB  MODFGS      *NO, RESET A FLAG
F9FA C4   FE        2     467         ANDB #%11111110  *     MODFGS(0)=0
F9FC F7   D041      5     468         STB  MODFGS
F9FF 85   02        2     469  SWB    BITA #%00000010
FA01 27   09        3     470         BEQ  NOTSWB      *SWITCH B SET?
FA03 F6   D041      5     471         LDB  MODFGS      *YES, SET B FLAG
FA06 CA   02        2     472         ORB  #%00000010  *     MODFGS(1)=1
FA08 F7   D041      5     473         STB  MODFGS
FA0B 39             5     474         RTS              *AND STOP
FA0C F6   D041      5     475  NOTSWB LDB  MODFGS      *NO, RESET B FLAG
FA0F C4   FD        2     476         ANDB #%11111101  *     MODFGS(1)=0
FA11 F7   D041      5     477         STB  MODFGS
FA14 85   04        2     478  SWC    BITA #%00000100
FA16 27   09        3     479         BEQ  NOTSWC      *SWITCH C SET?
FA18 F6   D041      5     480         LDB  MODFGS      *YES, SET C FLAG
FA1B CA   04        2     481         ORB  #%00000100  *     MODFGS(2)=1
FA1D F7   D041      5     482         STB  MODFGS
FA20 39             5     483         RTS              *AND STOP
```

```
FA21 F6   D041    5   485  NOTSWC LDB  MODFGS      *NO, RESET C FLAG
FA24 C4   FB      2   486         ANDB #%11111011 *    MODFGS(2)=0
FA26 F7   D041    5   487         STB  MODFGS
FA29 85   08      2   488  SWD    BITA #%00001000
FA2B 27   09      3   489         BEQ  NOTSWD     *SWITCH D SET?
FA2D F6   D041    5   490         LDB  MODFGS     *YES, SET D FLAG
FA30 CA   08      2   491         ORB  #%00001000 *     MODFGS(2)=0
FA32 F7   D041    5   492         STB  MODFGS
FA35 39           5   493         RTS             *AND STOP
FA36 F6   D041    5   494  NOTSWD LDB  MODFGS     *NO, RESET D FLAG
FA39 C4   F7      2   495         ANDB #%11110111 *    MODFGS(3)=0
FA3B F7   D041    5   496         STB  MODFGS
FA3E 39           5   497         RTS             *FINISHED
```

```
                          499  *  <<  SUBROUTINE ADJUST>>
                          500  *
                          501  *      CHECKS IF THE REEL OR CUTTER BAR RAISE OR LOWER SWITCHES ARE
                          502  *      PRESSED AND SETS THE CORRESPONDING FLAGS IN MODFGS
                          503  *      THE A AND B REGISTERS ARE MODIFIED BY THIS ROUTINE
                          504  *
FA3F F6   D041      5     505  ADJUST LDB   MODFGS       *GET THE SWITCH STATUS
FA42 B6   8000      5     506         LDA   PIASWD       *GET THE FLAGS STATUS
FA45 43             2     507         COMA               *INVERT THE SWITCHES TO POSITIVE LOGIC
FA46 C5   40        2     508         BITB  #%01000000   *CB ACTIVE?
FA48 26   04        3     509         BNE   CHCBSW       *YES, THEN CHECK THE CB SWITCH
FA4A C5   20        2     510         BITB  #%00100000   *NO, THEN CHECK IF RL ACTIVE?
FA4C 26   16        3     511         BNE   CHRLSW       *AND IF RL ACTIVE CHECK THE RL SWITCH
FA4E 85   40        2     512  CHCBSW BITA  #%01000000   *CB UP?
FA50 27   06        3     513         BEQ   CBDOWN       *NO
FA52 C4   DF        2     514         ANDB  #%11011111   *YES, MODFGS(5)=0
FA54 CA   50        2     515         ORB   #%01010000   *     MODFGS(4,6)=1
FA56 20   22        3     516         BRA   DONE
FA58 85   80        2     517  CBDOWN BITA  #%10000000   *CB DOWN?
FA5A 27   06        3     518         BEQ   NOCB         *NO
FA5C C4   CF        2     519         ANDB  #%11001111   *YES, MODFGS(4,5)=0
FA5E CA   40        2     520         ORB   #%01000000   *     MODFGS(6)=1
FA60 20   18        3     521         BRA   DONE
FA62 C4   BF        2     522  NOCB   ANDB  #%10111111   *CB INACTIVE, MODFGS(6)=0
FA64 85   10        2     523  CHRLSW BITA  #%00010000   *RL UP?
FA66 27   06        3     524         BEQ   RLDOWN       *NO
FA68 C4   BF        2     525         ANDB  #%10111111   *YES, MODFGS(6)=0
FA6A CA   30        2     526         ORB   #%00110000         MODFGS(4,5)=1
FA6C 20   0C        3     527         BRA   DONE
FA6E 85   20        2     528  RLDOWN BITA  #%00100000   *RL DOWN?
FA70 27   06        3     529         BEQ   NORL         *NO
FA72 C4   AF        2     530         ANDB  #%10101111   *YES, MODFGS(4,6)=0
FA74 CA   20        2     531         ORB   #%00100000         MODFGS(5)=1
FA76 20   02        3     532         BRA   DONE
FA78 C4   DF        2     533  NORL   ANDB  #%11011111   *RL INACTIVE, MODFGS(5)=0
FA7A F7   D041      5     534  DONE   STB   MODFGS       *SAVE THE UPDATED STATUS
FA7D 39             5     535         RTS
```

```
                          537  *  <<  SUBROUTINE DISPLAY >>
                          538  *
                          539  *      SUBROUTINE DISPLY UPDATES THE DISBUF REGISTERS
                          540  *
FA7E 108E D050      4     541  DISPLY LDY  #DISBUF     *BLANK THE DISPLAY
FA82 5F             2     542         CLRB
FA83 B6   20        2     543         LDA  #BLANK
FA85 A7   A5        5     544  LOOPDS STA  B,Y
FA87 5C             2     545         INCB
FA88 C1   10        2     546         CMPB #16         *B=16?
FA8A 26   F9        3     547         BNE  LOOPDS      *NO, THEN CONTINUE BLANKING
FA8C 7D   D041      7     548         TST  MODFGS      *AUTOFLAG?
FA8F 2B   2A        3     549         BMI  AUT1        *YES
FA91 77   D070      7     550         ASR REELAM       *REEL SWITCH IN AUTO?
FA94 24   0C        3     551         BCC MAN3         *NO
FA96 B6   D041      5     552         LDA MODFGS       *IS THE REEL SWITCH ACTIVE?
FA99 B5   20        2     553         BITA #%00100000
FA9B 27   05        3     554         BEQ MAN3         *IF NOT, THEN WRITE MAN
FA9D BD   FB27      8     555         JSR WRSET        *ELSE, WRITE SET
FAA0 20   03        3     556         BRA BACK1
FAA2 BD   FB17      8     557  MAN3   JSR  WRMAN       *NO, WRITE 'MAN_____'
FAA5 108E D005      4     558  BACK1  LDY  #ACBACT
FAA9 BD   FBC6      8     559         JSR  GET
FAAC 5D             2     560         TSTB             *B=$FF
FAAD 2A   08        3     561         BPL  NOFF1       *NO
FAAF 108E D005      4     562  MAN5   LDY  #ACBACT     *YES
FAB3 BD   FB66      8     563         JSR  WRCBAL      *WRITE ALL THE ACTUAL CB VALUES
FAB6 39             5     564         RTS              *AND STOP
FAB7 BD   FB47      8     565  NOFF1  JSR  WRCB        *WRITE THE ACTUAL CB VALUE
FABA 39             5     566  MAN4   RTS              *AND STOP
FABB B6   D041      5     567  AUT1   LDA  MODFGS
FABE 85   60        2     568         BITA #%01100000 *CB OR RL ACTIVE?
FAC0 27   32        3     569         BEQ  INACT       *NO
FAC2 BD   FB27      8     570         JSR  WRSET       *YES, WRITE 'SET_____'
FAC5 108E D006      4     571         LDY  #ACBSP
FAC9 BD   FBC6      8     572         JSR  GET
FACC 5D             2     573         TSTB             *B=$FF
FACD 2A   17        3     574         BPL  NOFF2       *NO
FACF B6   D041      5     575         LDA  MODFGS      *YES
FAD2 85   40        2     576         BITA #%01000000 *CB ACTIVE?
FAD4 27   08        3     577         BEQ  NOCBD       *NO
FAD6 108E D006      4     578         LDY  #ACBSP      *YES,
FADA BD   FB66      8     579         JSR  WRCBAL      *   WRITE ALL THE CB SET POINTS
FADD 39             5     580         RTS              *   AND STOP
```

```
FADE 108E D002     4   582  NOCBD  LDY  #ARSP
FAE2 BD   FB9C     8   583         JSR  WRRLAL     *WRITE ALL THE RL SETPOINTS
FAE5 39            5   584         RTS             *   AND STOP
FAE6 BD   FB47     8   585  NOFF2  JSR  WRCB       *WRITE THE SINGLE CB SETPOINT
FAE9 108E D002     4   586         LDY  #ARSP
FAED BD   FBC6     8   587         JSR  GET
FAF0 BD   FB58     8   588         JSR  WRRL       *WRITE THE SINGLE RL SETPOINT
FAF3 39            5   589         RTS             *   AND STOP
FAF4 BD   FB37     8   590  INACT  JSR  WRAUT
FAF7 108E D005     4   591         LDY  #ACBACT
FAFB BD   FBC6     8   592         JSR  GET
FAFE 5D            2   593         TSTB            *B=$FF?
FAFF 2A   08       3   594         BPL  NOFF3      *NO
FB01 108E D005     4   595         LDY  #ACBACT    *YES, WRITE ALL THE ACTUAL CB VALUES
FB05 BD   FB66     8   596         JSR  WRCBAL
FB08 39            5   597         RTS             *AND STOP
FB09 BD   FB47     8   598  NOFF3  JSR  WRCB       *WRITE THE ACTUAL CB VALUE
FB0C 108E D001     4   599         LDY  #ARACT     *WRITE THE ACTUAL RL VALUE
FB10 BD   FBC6     8   600         JSR  GET
FB13 BD   FB58     8   601         JSR  WRRL
FB16 39            5.  602         RTS             *AND STOP
```

```
                        604  *
                        605  *       THE FOLLOWING ROUTINES ARE USED BY DISPLY
                        606  *
FB17 86   4D       2    607  WRMAN   LDA  #MCHAR          *LOAD 'MAN_____________'
FB19 B7   D050     5    608         STA  DISBUF
FB1C 86   41       2    609         LDA  #ACHAR
FB1E B7   D051     5    610         STA  DISBUF+1
FB21 86   4E       2    611         LDA  #NCHAR
FB23 B7   D052     5    612         STA  DISBUF+2
FB26 39            5    613         RTS
FB27 86   53       2    614  WRSET   LDA  #SCHAR .        *LOAD 'SET_____________'
FB29 B7   D050     5    615         STA  DISBUF
FB2C 86   45       2    616         LDA  #ECHAR
FB2E B7   D051     5    617         STA  DISBUF+1
FB31 86   54       2    618         LDA  #TCHAR
FB33 B7   D052     5    619         STA  DISBUF+2
FB36 39            5    620         RTS
FB37 86   41       2    621  WRAUT   LDA  #ACHAR      *LOAD 'AUT_____________'
FB39 B7   D050     5    622         STA  DISBUF
FB3C 86   55       2    623         LDA  #UCHAR
FB3E B7   D051     5    624         STA  DISBUF+1
FB41 86   54       2    625         LDA  #TCHAR
FB43 B7   D052     5    626         STA  DISBUF+2
FB46 39            5    627         RTS
FB47 F7   D059     5    628  WRCB    STB  DISBUF+9    *LOAD '______CB**___'
FB4A B7   D058     5    629         STA  DISBUF+8
FB4D C6   43       2    630         LDB  #CCHAR
FB4F F7   D056     5    631         STB  DISBUF+6
FB52 C6   42       2    632         LDB  #BCHAR
FB54 F7   D057     5    633         STB  DISBUF+7
FB57 39            5    634         RTS
FB58 F7   D05D     5    635  WRRL    STB  DISBUF+13   *LOAD '____________RL*_'
FB5B C6   52       2    636         LDB  #RCHAR
FB5D F7   D05B     5    637         STB  DISBUF+11
FB60 C6   4C       2    638         LDB  #LCHAR
FB62 F7   D05C     5    639         STB  DISBUF+12
FB65 39            5    640         RTS
```

```
                          642  *  <<  SUBROUTINE WRCBAL >>
                          643  *      WRITES ALL THE VALUES FOR A THROUGH D
                          644  *      GIVEN THE STARTING ADDRESS OF THEM IN THE Y REGISTER
                          645  *      THE ROUTINE SPLIT IS CALLED
                          646  *      TO CONVERT THE NUMBERS TO ASCII CODE
FB66 E6   A4        4     647  WRCBAL LDB  ,Y          *WRITE A
FB68 BD   FC00      8     648         JSR  SPLIT
FB6B F7   D056      5     649         STB  DISBUF+6
FB6E B7   D055      5     650         STA  DISBUF+5
FB71 31   A8 10     5     651         LEAY $10,Y      *ADDRESS THE NEXT VALUE
FB74 E6   A4        4     652         LDB  ,Y         *WRITE B
FB76 BD   FC00      8     653         JSR  SPLIT
FB79 F7   D059      5     654         STB  DISBUF+9
FB7C B7   D058      5     655         STA  DISBUF+8
FB7F 31   A8 10     5     656         LEAY $10,Y      *ADDRESS THE NEXT VALUE
FB82 E6   A4        4     657         LDB  ,Y         *WRITE C
FB84 BD   FC00      8     658         JSR  SPLIT
FB87 F7   D05C      5     659         STB  DISBUF+12
FB8A B7   D05B      5     660         STA  DISBUF+11
FB8D 31   A8 10     5     661         LEAY $10,Y      *ADDRESS THE NEXT VALUE
FB90 E6   A4        4     662         LDB  ,Y         *WRITE D
FB92 BD   FC00      8     663         JSR  SPLIT
FB95 F7   D05F      5     664         STB  DISBUF+15
FB98 B7   D05E      5     665         STA  DISBUF+14
FB9B 39             5     666         RTS             *DONE
                          667  *
                          668  *  <<  SUBROUTINE WRRLAL >>
                          669  *      WRITES ALL THE VALUES FOR A THROUGH D
                          670  *      GIVEN THE STARTING ADDRESS OF THEM IN THE Y REGISTER
                          671  *      THE ROUTINE SPLIT IS CALLED
                          672  *      TO CONVERT THE NUMBERS TO ASCII CODE
FB9C E6   A4        4     673  WRRLAL LDB  ,Y          *WRITE A
FB9E BD   FC00      8     674         JSR  SPLIT
FBA1 F7   D055      5     675         STB  DISBUF+5
FBA4 31   A8 10     5     676         LEAY $10,Y      *ADDRESS THE NEXT VALUE
FBA7 E6   A4        4     677         LDB  ,Y         *WRITE B
FBA9 BD   FC00      8     678         JSR  SPLIT
FBAC F7   D058      5     679         STB  DISBUF+8
FBAF 31   A8 10     5     680         LEAY $10,Y      *ADDRESS THE NEXT VALUE
FBB2 E6   A4        4     681         LDB  ,Y         *WRITE C
FBB4 BD   FC00      8     682         JSR  SPLIT
FBB7 F7   D05B      5     683         STB  DISBUF+11
FBBA 31   A8 10     5     684         LEAY $10,Y      *ADDRESS THE NEXT VALUE
FBBD E6   A4        4     685         LDB  ,Y         *WRITE  D
FBBF BD   FC00      8     686         JSR  SPLIT
FBC2 F7   D05E      5     687         STB  DISBUF+14
FBC5 39             5     688         RTS             *DONE
```

```
                          690  *
                          691  *   <<  SUBROUTINE GET>>
                          692  *
                          693  *       GIVEN THE STARTING ADDRESS OF THE A ACTUAL OR SETPOINT
                          694  *       IN THE Y REGISTER
                          695  *       THE ROUTINE EXAMINES THE MODFGS AND RETURNS THE
                          696  *       ASCII CODE FOR THE VALUE WHOSE FLAG WAS SET
                          697  *       IN THE A AND B ACCUMULATORS
                          698  *       IF NO FLAGS A THROUGH D ARE SET THE ROUTINE RETURNS
                          699  *       $FF IN THE B REGISTER
                          700  *       IN ADDITION IF NO FLAGS WERE SET THE LETTERS A,B,C,D
                          701  *       ARE WRITTEN TO THE DISBUF REGISTERS
                          702  *
FBC6 86   41        2     703  GET     LDA  #ACHAR       *LOAD 'A' INTO CHRBUF
FBC8 B7   D045      5     704          STA  CHRBUF
FBCB B6   D041      5     705          LDA  MODFGS       *GET THE FLAGS STATUS
FBCE 5F             2     706          CLRB
FBCF 47             2     707  LOOPG   ASRA              *SHIFT A RIGHT
FBD0 25   22        3     708          BCS  FOUND
FBD2 31   A8 10     5     709          LEAY $10,Y        *ADVANCE THE INDEX TO THE NEXT VALUE
FBD5 7C   D045      7     710          INC  CHRBUF       *ADVANCE THE CHARACTER VALUE
FBD8 5C             2     711          INCB              *INCREMENT THE COUNT
FBD9 C1   04        2     712          CMPB #4           *COUNT=4?
FBDB 26   F2        3     713          BNE  LOOPG        *NO, CONTINUE LOOPING
FBDD 86   41        2     714          LDA  #ACHAR       *NO VALUE FOUND
FBDF B7   D054      5     715          STA  DISBUF+4     *WRITE '____A__B__C__D__'
FBE2 86   42        2     716          LDA  #BCHAR
FBE4 B7   D057      5     717          STA  DISBUF+7
FBE7 86   43        2     718          LDA  #CCHAR
FBE9 B7   D05A      5     719          STA  DISBUF+10
FBEC 86   44        2     720          LDA  #DCHAR
FBEE B7   D05D      5     721          STA  DISBUF+13
FBF1 C6   FF        2     722          LDB  #$FF         *AND SET THE FLAG
FBF3 39             5     723          RTS               *THEN DONE
FBF4 B6   D045      5     724  FOUND   LDA  CHRBUF       *GET THE CHARACTER WHOSE FLAG WAS SET
FBF7 B7   D054      5     725          STA  DISBUF+4     *WRITE THE CHARACTER TO THE DISPLAY
FBFA E6   A4        4     726          LDB  ,Y           *GET THE VALUE FOR WHOSE FLAG WAS SET
FBFC BD   FC00      8     727          JSR  SPLIT
FBFF 39             5     728          RTS
```

```
                          730  *
                          731  *       SUBROUTINE SPLIT
                          732  *
                          733  *       TAKES THE BCD NUMBER IN THE B REGISTER AND
                          734  *       RETURNS THE CORRESPONDING ASCII CODE IN REGISTERS A & B
                          735  *       THE HIGH ORDER NIBBLE IS RETURNED IN A AND THE LOW
                          736  *       ORDER NIBBLE IN B
                          737  *       THE MUL OPERATION IS USED TO SHIFT B LEFT FOUR TIMES INTO A
                          738  *
FC00 86   10         2    739  SPLIT   LDA  #16
FC02 3D              11   740          MUL              *SHIFT B 4 TIMES LEFT INTO A
FC03 54              2    741          LSRB             *SHIFT B RIGHT 4 TIMES
FC04 54              2    742          LSRB
FC05 54              2    743          LSRB
FC06 54              2    744          LSRB
FC07 8B   30         2    745          ADDA #$30        *CONVERT TO ASCII CODE
FC09 CB   30         2    746          ADDB #$30        *CONVERT TO ASCII CODE
FC0B 39              5    747          RTS
```

```
                        749  *  << LCD DISPLAY RFRSH ROUTINE >>
                        750  *
                        751  *       EACH OF THE 16 DISPLAY CHARACTERS ARE CHECKED WITH THE
                        752  *       REFRESH BUFFER AND REPLACED IF DIFFERENT
                        753  *
FC0C 36    10       5   754  RFRSH     PSHU X              *STORE CURRENT IX REGISTER
FC0E B6    8002     5   755            LDA ATPIAD          *ENABLE THE CONSOLE DATA BUS
FC11 84    BF       2   756            ANDA #%10111111
FC13 B7    8002     5   757            STA ATPIAD
FC16 8E    D050     3   758            LDX #DISBUF         *LOAD ADDRESS OF REFRESH BUFFER
FC19 C6    10       2   759            LDB #16             *SET BUFFER COUNTER
FC1B 86    84       2   760  NXTCHR    LDA #GETCHR         *READ CHARACTER FROM LCD
FC1D 97    00       4   761            STA LCD
FC1F 96    00       4   762            LDA LCD
FC21 A1    80       6   763            CMPA ,X+            *CHECK IF LCD CHAR. = DISBUF CHARACTER
FC23 27    07       3   764            BEQ SAME            *YES
FC25 A6    1F       5   765            LDA -1,X            *NO, LOAD LCD WITH NEW CHARACTER
FC27 97    00       4   766            STA LCD
FC29 BD    FC4D     8   767            JSR SDELAY          *WAIT FOR CHARACTER TO LOAD
FC2C 86    89       2   768  SAME      LDA #INCCSR         *INC CURSOR FOR THE NEXT CHAR. COMPARISON
FC2E 97    00       4   769            STA LCD
FC30 5A             2   770            DEC B               *DECREMENT BUFFER COUNTER
FC31 5D             2   771            TSTB                *FINISHED?
FC32 26    E7       3   772            BNE NXTCHR          *NO: PROCESS NEXT CHARACTER
FC34 37    10       5   773            PULU X              *YES: RESTORE IX REGISTER
FC36 0F    00       6   774            CLR LCD             *SET CHARACTER CURSOR AT THE START
FC38 B6    8002     5   775            LDA ATPIAD          *DISABLE THE CONSOLE DATA BUS
FC3B 8A    40       2   776            ORA #%01000000
FC3D B7    8002     5   777            STA ATPIAD
FC40 39             5   778            RTS                 *THEN RETURN TO THE MAIN ROUTINE
```

```
                          780  *  << LONG DELAY SUBROUTINE >>
                          781  *
                          782  *     THIS SUBROUTINE IS USED AS A 3.5ms DELAY TIME WHICH IS USED
                          783  *     IN CONJUNCTION WITH THE LCD REFRESH ROUTINE TO ALLOW TIME
                          784  *     FOR THE CURSOR TO BE REPOSITIONED.
                          785  *
FC41 36    10        5    786  LDELAY   PSHU X            *STORE CURRENT IX VALUE
FC43 8E    0500      3    787         LDX #$500          *LOAD IX WITH 3.5ms COUNT
FC46 30    1F        5    788  LOOP    LEAX -1,X          *DEC IX
FC48 26    FC        3    789         BNE LOOP           *CONTINUE LOOPING IF IX NE O
FC4A 37    10        5    790         PULU X             *RESTORE OLD IX VALUE
FC4C 39              5    791         RTS
                          792  *
                          793  *  << SHORT DELAY SUBROUTINE >>
                          794  *
                          795  *       THIS SUBROUTINE IS USED AS A .8ms DELAY TIME WHICH IS USED
                          796  *       IN CONJUNCTION WITH THE LCD REFRESH ROUTINE TO ALLOW A
                          797  *       CHARACTER TO BE LATCHED INTO THE LCD DISPLAY.
                          798  *
FC4D 36    10        5    799  SDELAY   PSHU X            *STORE CURRENT IX VALUE
FC4F 8E    0050      3    800         LDX #$50           *LOAD IX WITH .8ms COUNT
FC52 30    1F        5    801  LOOP1   LEAX -1,X          *DEC IX
FC54 26    FC        3    802         BNE LOOP1          *CONTINUE LOOPING IF IX NE O
FC56 37    10        5    803         PULU X             *RESTORE OLD IX VALUE
FC58 39              5    804         RTS
```

```
                        806  *  << SUBROUTINE COMPAR >>
                        807  *
                        808  *       THIS ROUTINE DETERMINES IF AUTOMATIC OR MANUAL
                        809  *       CONTROL IS TO BE INITIATED
                        810  *
FC59 86   02       2    811  COMPAR   LDA #02           *FIRST SET OFFSET
FC5B B7   D072     5    812           STA OFFSET
FC5E 7C   D07A     7    813          INC SCOUNT         *KEEP TRACK OF THE NUMBER OF DATUM SENT
FC61 B6   D041     5    814  OVER8    LDA MODFGS         *MODIFY MODFGS INTO ALLFGS
FC64 85   0F       2    815           BITA #%00001111    *IF NO SWITCHES ARE PRESSED,
FC66 26   02       3    816           BNE PASS           *THEN SET ALL SWITCH BITS HIGH,
FC68 8A   0F       2    817           ORA #%00001111     *SO ALL SENSOR SETS ACT AS ONE
FC6A B7   D046     5    818  PASS     STA ALLFGS
FC6D B6   D046     5    819           LDA ALLFGS         *CHECK MODFGS TO DETERMINE STATE
FC70 4D            2    820           TSTA
FC71 2A   05       3    821           BPL MN
FC73 BD   FC7C     8    822           JSR AUTO           *JSR AUTO IF AUTO FLAG IS SET
FC76 20   03       3    823           BRA BACK
FC78 BD   FEC7     8    824  MN       JSR MANUAL         *JSR MANUAL IF MANUAL FLAG IS SET
FC7B 39            5    825  BACK     RTS                *RETURN TO THE MAIN ROUTINE
```

```
                              827  *  << SUBROUTINE AUTO >>
                              828  *
                              829  *THIS SUBROUTINE INITIATES AUTO CONTROL BY PROCESSING
                              830  *       THE SENSOR INFORMATION COLLECTED IN POLCR
                              831  *
FC7C 86   01         2        832  AUTO      LDA #$01          *INITIALIZE MODPQS=1
FC7E B7   D043       5        833            STA MODPOS
FC81 8E   D000       3        834            LDX #$D000        *LOAD DATA MEMORY BASE ADDRESS INTO X
                              835  *
                              836  *       FIRST PROCESS REEL DATA AND CONTROL
                              837  *
FC84 BD   FCEB       8        838  AUNEXT    JSR RCALC         *PROCESS REEL DATA AND CHECK FOR ERROR
FC87 4D                      2  839            TSTA              *CHECK ERROR FLAG
FC88 2B   0A         3        840            BMI RERROR        *YES AN ERROR EXISTS GOTO RERROR
FC8A 7D   D070       7        841            TST REELAM        *REEL SWITCH IN AUTO POSITION?
FC8D 27   05         3        842            BEQ RERROR       *IF IN MANUAL, GO TO RMANUAL ROUTINE
FC8F BD   FD11       8        843            JSR RCMPAR        *NO ERROR EXISTS INITIATE CONTROL
FC92 20   03         3        844            BRA CB            *CONTINUE TO CB DATA AND CONTROL BLOCK
FC94 BD   FF18       8        845  RERROR    JSR RMANUL        *INITIATE MANUAL CONTROL SPEC. BY MODFGS
                              846  *
                              847  *       SECONDLY, PROCESS CB DATA AND CONTROL
                              848  *
FC97 A6   07         5        849  CB        LDA 7,X           *CHECK CB ERROR STATUS
FC99 4D                      2  850            TSTA
FC9A 27   08         3        851            BEQ CBERR         *BRANCH TO CBERR IF AN ERROR OCCURRED
FC9C BD   FD9D       8        852            JSR CBCALC        *CALCULATE THE CB HEIGHT
FC9F BD   FE4C       8        853            JSR CBCMPR        *INITIATE CB CONTROL
FCA2 20   07         3        854            BRA OUT           *FINISHED AUTO CB CONTROL
FCA4 86   FF         2        855  CBERR     LDA #$FF          *IF CB DATA IS INVALID LOAD ERROR
FCA6 A7   05         5        856            STA 5,X           *FLAG INTO *CBACT
FCA8 BD   FF4F       8        857            JSR CBMAN         *INITIATE MANUAL CONTROL
FCAB E6   05         5        858  OUT       LDB 5,X           *CHANGE THE ACTUAL NUMBERS TO ASC11
FCAD BD   FC00       8        859            JSR SPLIT         *CHARACTERS AND STORE THEN IN RAM
FCB0 A7   89 0082    8        860            STA $82,X         *SO THEY CAN BE SENT OUT TO TAPE STORAGE
FCB4 E7   89 0083    8        861            STB $83,X
FCB8 86   2C         2        862            LDA #$2C          *SEPARATE DATA WITH A COMMA
FCBA A7   89 0084    8        863            STA $84,X
FCBE 6F   03         7        864            CLR 3,X           *CLEAR CB SUM REGISTERS
FCC0 6F   04         7        865            CLR 4,X
FCC2 6F   0A         7        866            CLR 10,X
FCC4 6F   0B         7        867            CLR 11,X
FCC6 6F   0E         7        868            CLR 14,X
FCC8 6F   0F         7        869            CLR 15,X
```

```
FCCA B6   08      2   871   LDA #08
FCCC A7   07      5   872   STA 7,X
FCCE 30   88 10   5   873   LEAX $10,X      *INC X TO PROCESS THE NEXT SENSOR SET
FCD1 78   D043    7   874   LSL MODPOS       *SET SENSOR SET POINTER FOR MODFGS
FCD4 8C   D040    4   875   CMPX #$D040     *FINISHED A, B, C, AND D SENSOR SETS?
FCD7 26   AB      3   876   BNE AUNEXT      *IF NO PROCESS NEXT SENSOR SET
FCD9 B6   D092    5   877   LDA $D092       MOVE THE B SENSOR DATA
FCDC B7   D085    5   878   STA $D085
FCDF B6   D093    5   879   LDA $D093
FCE2 B7   D086    5   880   STA $D086
FCE5 86   20      2   881   LDA #BLANK
FCE7 B7   D087    5   882   STA $D087
FCEA 39           5   883   RTS             *IF FINISHED RETURN TO SUB COMPARE
```

```
                                885  *  <<  SUBROUTINE RCALC >>
                                886  *
                                887  *       THIS SUBROUTINE WORKS IN CONJUNCTION WITH THE AUTO
                                888  *       SUBROUTINE.  THE REEL SUM REGISTER IS DIVIDED BY
                                889  *       EIGHT, ERROR CHECKED, THE RESULT THEN SAVED IN *RACT
                                890  *       AND PASSED TO AUTO IN THE A REGISTER
                                891  *
FCEB A6   84        4           892  RCALC     LDA ,X              *LOAD THE REEL SUM
FCED 2B   08        3           893            BMI FIXERR          *ERROR IF REEL SUM >127
FCEF 44             2           894            LSRA                *DIVIDE BY 8
FCF0 44             2           895            LSRA
FCF1 44             2           896            LSRA
FCF2 81   04        2           897            CMPA #$04           *ERROR IF REEL VALUE > 4
FCF4 2E   04        3           898            BGT FIXERR
FCF6 84   0F        2           899            ANDA #%00001111     *MASK HIGH NIBBLE
FCF8 20   02        3           900            BRA ENDCAL          *FINISHED WITH CORRECT REEL VALUE
FCFA 86   FF        2           901  FIXERR    LDA #$FF            *LOAD ERROR FLAG
FCFC A7   01        5           902  ENDCAL    STA 1,X             *STORE RESULT IN *RACT REGISTER
FCFE 84   0F        2           903            ANDA #%00001111     *CALCULATE THE ASCII VALUE OF RACT
FD00 8B   30        2           904            ADDA #$30
FD02 A7   89 0085   8           905            STA $85,X           *AND STORE IT IN RAM
FD06 86   0D        2           906            LDA #$0D            *THEN ADD A BLANK
FD08 A7   89 0086   8           907            STA $86,X
FD0C 6F   84        6           908            CLR  ,X             *CLEAR SUM REGISTER
FD0E A6   01        5           909            LDA 1,X
FD10 39             5           910            RTS                 *RETURN TO SUB AUTO
```

```
                          912  *  <<  SUBROUTINE RCMPAR >>
                          913  *
                          914  *        THIS SUBROUTINE UPDATES SETPOINT CHANGES AND
                          915  *        INITIATES CONTROL AS NECESSARY. RCMPAR IS CALLED
                          916  *        AND RETURNED BY SUB AUTO
                          917  *
FD11 B6   D046       5    918  RCMPAR    LDA ALLFGS
FD14 85   20         2    919            BITA #%00100000    *CHECK IF REEL SETPOINT IS TO BE CHANGED
FD16 27   0A         3    920            BEQ NOSPCH         *SKIP SETPOINT CHANGE IF FLAG RESET
FD18 86   03         2    921            LDA #$03           *SET SPLIM TO 3
FD1A B7   D042       5    922            STA SPLIM
FD1D C6   02         2    923            LDB #$02           *LOAD B ACC. INDEXED OFFSET
FD1F BD   FD74       8    924            JSR SPCHNG         *JUMP TO SP CHANGING ROUTINE
                          925  *
                          926  *        INITIATE CONTROL OF REEL HYDRAULIC ACTUATORS
                          927  *
FD22 A6   01         5    928  NOSPCH    LDA 1,X            *LOAD THE ACTUAL REEL VALUE *RACT
FD24 A1   02         5    929            CMPA 2,X           *COMPARE ACTUAL AND SETPOINT VALUES
FD26 27   3F         3    930            BEQ NOCTL          *SP=ACT NO CONTROL
FD28 2D   1F         3    931            BLT LOWER          *ACT>SP RAISE REEL
FD2A A6   89 00C0    8    932           LDA $C0,X           *PULSE THE REEL UP
FD2E 4D              2    933           TSTA
FD2F 27   06         3    934           BEQ RAISE
FD31 6A   89 00C0   10    935           DEC $C0,X
FD35 20   32         3    936           BRA NOCNTL
FD37 6C   89 00C0   10    937  RAISE    INC $C0,X
FD3B A6   89 F000    8    938            LDA -$1000,X       *ACT<SP LOWER REEL
FD3F 84   D0         2    939            ANDA #%11010000    *SET RAISE BIT ON PIA
FD41 8A   10         2    940            ORA #%00010000
FD43 A7   89 F000    8    941            STA -$1000,X
FD47 20   2A         3    942            BRA ENDRCM         *GO TO END
FD49 6D   89 00C0   10    943  LOWER     TST $C0,X          *PULSE THE REEL DOWN
FD4D 27   06         3    944            BEQ LOWER1
FD4F 6A   89 00C0   10    945            DEC $C0,X
FD53 20   14         3    946            BRA NOCNTL
FD55 6C   89 00C0   10    947  LOWER1    INC $C0,X
FD59 A6   89 F000    8    948            LDA -$1000,X       *SET LOWER BIT ON PIA
FD5D 84   E0         2    949            ANDA #%11100000
FD5F 8A   20         2    950            ORA #%00100000
FD61 A7   89 F000    8    951            STA -$1000,X
FD65 20   0C         3    952            BRA ENDRCM         *GOTO END
FD67 6F   0D         7    953  NOCTL       CLR 13,X        *IF NO CONTROL CLEAR PULSE FLAG
FD69 A6   89 F000    8    954  NOCNTL    LDA -$1000,X       *SET BOTH RAISE AND LOWER BITS OFF
FD6D 84   C0         2    955            ANDA #%11000000
FD6F A7   89 F000    8    956            STA -$1000,X
FD73 39              5    957  ENDRCM    RTS                *RETURN TO SUB AUTO
```

```
                        959  *  <<  SUBROUTINE SPCHNG >>
                        960  *
                        961  *       THIS SUBROUTINE CHANGES THE SETPOINT ACCORDING TO
                        962  *       ALLFGS.  THIS ROUTINE IS CALLED BY RCMPAR AND CBCMPR
                        963  *
FD74 B6   D046      5   964  SPCHNG   LDA ALLFGS
FD77 B5   D043      5   965          BITA MODPOS          *CHANGE SETPOINT?
FD7A 27   20        3   966          BEQ ENDSP            *NO QUIT
FD7C 85   10        2   967          BITA #%00010000      *INCREMENT SETPOINT?
FD7E 27   10        3   968          BEQ LSP              *NO LOWER SETPOINT
FD80 A6   85        5   969          LDA B,X              *LOAD SETPOINT
FD82 B1   D042      5   970          CMPA SPLIM           *CHECK IF SP IS AT MAX LIMIT
FD85 27   15        3   971          BEQ ENDSP            *IF SP IS AT MAX LIMIT, QUIT
FD87 A6   85        5   972          LDA B,X              *IF OK INCREMENT SETPOINT
FD89 8B   01        2   973          ADDA #1              *INCREMENT IN BCD NOTATION
FD8B 19             2   974          DAA
FD8C A7   85        5   975          STA B,X
FD8E 20   0C        3   976          BRA ENDSP            *FINISHED RAISING SETPOINT QUIT
FD90 A6   85        5   977  LSP     LDA B,X              *LOWER THE SETPOINT
FD92 4D             2   978          TSTA                 *SETPOINT AT LOW LIMIT ?
FD93 27   07        3   979          BEQ ENDSP            *IF YES THEN QUIT
FD95 A6   85        5   980          LDA B,X              *ELSE DECREMENT SP.
FD97 8B   99        2   981          ADDA #$99            *DECREMENT IN BCD NOTATION
FD99 19             2   982          DAA
FD9A A7   85        5   983          STA B,X
FD9C 39             5   984  ENDSP   RTS                  *RETURN TO EITHER RCMPAR OR CBCMPR
```

```
                        986  *  <<  SUBROUTINE CBCALC >>
                        987  *
                        988  *       THIS ROUTINE CALCULATES THE CB HEIGHT FROM THE FIRST
                        989  *       AND SECOND ECHO REGISTERS
                        990  *
FD9D EC   03       6    991  CBCALC   LDD 3,X          *DIVIDE THE FIRST ECHO BY
FD9F BD   FDE5     8    992          JSR DIV16        *16 AND RESTORE IT
FDA2 ED   03       6    993          STD 3,X
FDA4 EC   0E       6    994          LDD 14,X         *DIVIDE THE SECOND ECHO BY
FDA6 BD   FDE5     8    995          JSR DIV16        *16, ADD THE FIRST ECHO TO IT,
FDA9 E3   03       7    996          ADDD 3,X         *AND RESTORE THE RESULT AS THE
FDAB ED   0E       6    997          STD 14,X         *DIVIDEND
FDAD EC   03       6    998          LDD 3,X          *DIVIDE THE FIRST ECHO AGAIN BY
FDAF BD   FDE5     8    999          JSR DIV16        *16, AND RESTORE THE RESULT
FDB2 ED   03       6   1000          STD 3,X          *AS THE DIVISOR
                       1001  *
                       1002  *       DIVISION ROUTINE
                       1003  *
                       1004  *       THIS ROUTINE DIVIDES A 16 BIT UNSIGNED DIVIDEND BY A 8
                       1005  *       BIT DIVISOR
                       1006  *
FDB4 86   08       2   1007          LDA #8           *LOAD THE COUNT VALUE
FDB6 A7   08       5   1008          STA 8,X
FDB8 EC   0E       6   1009          LDD 14,X         *LOAD THE DIVIDEND
FDBA 58            2   1010  DIVIDE   A$LB            *SHIFT THE DIVIDEND, QUOTIENT
FDBB 49            2   1011          ROLA
FDBC A1   04       5   1012          CMPA 4,X         *CHECK SUBTRACTION SUCCESS
FDBE 25   03       3   1013          BCS CHKCNT       *IF YES, SUBTRACT AND SET
FDC0 A0   04       5   1014          SUBA 4,X         *THE BIT IN QUOTIENT
FDC2 5C            2   1015          INCB
FDC3 6A   08       7   1016  CHKCNT   DEC 8,X
FDC5 26   F3       3   1017          BNE DIVIDE
FDC7 ED   08       6   1018          STD 8,X          *STORE THE REMAINDER AND QUOTIENT
```

```
                        1020  *
                        1021  *       NEXT MULTIPLY THE QUOTIENT BY TEN
                        1022  *
FDC9 4F            2    1023           CLRA
FDCA E6   09       5    1024           LDB 9,X            *LOAD THE QUOTIENT
FDCC 58            2    1025           LSLB               *MULTIPLY BY TWO
FDCD 49            2    1026           ROLA
FDCE ED   0A       6    1027           STD 10,X           *STORE FIRST MULTIPLICATION
FDDO 58            2    1028           LSLB               *THEN MULTIPLY BY 8
FDD1 49            2    1029           ROLA
FDD2 58            2    1030           LSLB
FDD3 49            2    1031           ROLA
FDD4 E3   0A       7    1032           ADDD 10,X          *THEN ADD THE FIRST MULTIPLICATION
FDD6 BD   FDE5     8    1033           JSR DIV16          *DIVIDE THE RESULT BY 16
FDD9 CO   0A       2    1034           SUBB #10           *SUBTRACT TEN
FDDB B6   DO44     5    1035           LDA PIPELH         *LOAD THE MULTIPLICATION CONSTANT
FDDE 3D           11    1036           MUL                *THEN MULTIPLY
FDDF ED   0C       6    1037           STD 12,X           *STORE THE RESULT AND
FDE1 BD   FDF1     8    1038           JSR BCD            *CHANGE HEX TO BCD
FDE4 39            5    1039           RTS                *RETURN TO AUTO
                        1040  *
                        1041  *   <<  SUBROUTINE DIV16 >>
                        1042  *
                        1043  *       THIS ROUTINE IS CALLED BY CBCALC AND DIVIDES
                        1044  *       THE DATA IN THE D REGISTER BY 16, AND RESTORES
                        1045  *       THE RESULT BACK IN THE D REGISTER.  THE ROUTINE
                        1046  *       ALSO ROUNDS THE CARRY TO THE NEAREST INTEGER VALUE
                        1047  *
FDE5 44            2    1048  DIV16    LSRA               *DIVIDE BY 2
FDE6 56            2    1049           RORB
FDE7 44            2    1050           LSRA               *DIVIDE BY 4
FDE8 56            2    1051           RORB
FDE9 44            2    1052           LSRA               *DIVIDE BY 8
FDEA 56            2    1053           RORB
FDEB 44            2    1054           LSRA               *DIVIDE BY 16
FDEC 56            2    1055           RORB
FDED 24   01       3    1056           BCC OKAY2          *IF CARRY CLEAR THEN END
FDEF 5C            2    1057           INCB               *OTHERWISE INCREMENT THE RESULT
FDFO 39            5    1058  OKAY2    RTS                *THEN RETURN TO CBCALC
```

```
                        1060  *
                        1061  *  <<  SUBROUTINE BCD >>
                        1062  *
                        1063  *       THIS ROUTINE IS CALLED BY ROUTINE CBCALC AND CONVERTS
                        1064  *       A 16 BIT BINARY NUMBER INTO BCD NOTATION USING THE
                        1065  *       ADD 3 ALGORITHM
                        1066  *
FDF1 6F    09       7   1067  BCD      CLR 9,X           *CLEAR TEMPORARY STORAGE AREAS
FDF3 6F    0A       7   1068           CLR 10,X
FDF5 86    10       2   1069           LDA #16           *LOAD THE 16 BIT COUNTER
FDF7 A7    0B       5   1070           STA 11,X
FDF9 6A    0B       7   1071  ROUND3   DEC 11,X          *EACH ROUND DEC THE COUNTER
FDFB 68    0D       7   1072           ASL 13,X          *SHIFT LEFT THE LOW BINARY BYTE
FDFD 69    0C       7   1073           ROL 12,X          *SHIFT LEFT THE HIGH BINARY BYTE
FDFF 69    0A       7   1074           ROL 10,X          *SHIFT LEFT THE LOW TEMP BCD STORAGE
FE01 69    09       7   1075           ROL 9,X           *SHIFT LEFT THE HIGH TEMP BCD STORAGE
FE03 6D    0B       7   1076           TST 11,X          *ALL 16 BITS SHIFTED ?
FE05 27    36       3   1077           BEQ FINN2         *IF YES THEN END
FE07 EC    09       6   1078           LDD 9,X           *LOAD THE TEMP BCD MEMORY
FE09 C4    0F       2   1079           ANDB #%00001111   *CHECK IF LOW NIBBLE, LOW BYTE >= 5
FE0B C1    05       2   1080           CMPB #5
FE0D 25    06       3   1081           BLO OKAY3         *IF NOT CONTINUE TO NEXT TEST
FE0F E6    0A       5   1082           LDB 10,X          *OTHERWISE ADD 3 TO THE LOW NIBBLE
FE11 CB    03       2   1083           ADDB #3
FE13 E7    0A       5   1084           STB 10,X
FE15 84    0F       2   1085  OKAY3    ANDA #%00001111   *CHECK IF LOW NIBBLE, HIGH BYTE >= 5
FE17 81    05       2   1086           CMPA #5
FE19 25    06       3   1087           BLO OKAY4         *IF NOT CONTINUE TO THE NEXT TEST
FE1B A6    09       5   1088           LDA 9,X           *OTHERWISE ADD 3 TO THE LOW NIBBLE
FE1D 8B    03       2   1089           ADDA #3
FE1F A7    09       5   1090           STA 9,X
FE21 EC    09       6   1091  OKAY4    LDD 9,X           *RELOAD TEMP BCD TO CHECK HIGH NIBBLES
FE23 C4    F0       2   1092           ANDB #%11110000   *CHECK IF HIGH NIBBLE, LOW BYTE >= 5
FE25 C1    50       2   1093           CMPB #$50
FE27 25    06       3   1094           BLO OKAY5         *IF NOT CONTINUE TO THE NEXT TEST
FE29 E6    0A       5   1095           LDB 10,X          *OTHERWISE ADD 3 TO THE HIGH NIBBLE
FE2B CB    30       2   1096           ADDB #$30
FE2D E7    0A       5   1097           STB 10,X
```

```
FE2F 84  F0      2  1099  OKAY5   ANDA #%11110000   *CHECK IF HIGH NIBBLE, HIGH BYTE >= 5
FE31 81  50      2  1100          CMPA #$50
FE33 25  06      3  1101          BLO OKAY6         *IF NOT CONTINUE TO END
FE35 A6  09      5  1102          LDA 9,X           *OTHERWISE ADD 3 TO THE HIGH NIBBLE
FE37 8B  30      2  1103          ADDA #$30
FE39 A7  09      5  1104          STA 9,X
FE3B 20  BC      3  1105  OKAY6   BRA ROUND3        *ALL 16 BITS SHIFTED ?,IF NOT RELOOP
FE3D A6  09      5  1106  FINN2   LDA 9,X           *ELSE SUBTRACT AN OFFSET OF 5
FE3F 81  05      2  1107          CMPA #5           *IF THE NUMBER IS LESS THAN 5
FE41 2C  03      3  1108          BGE OKAY7         *THEN SET IT TO ZERO
FE43 4F          2  1109          CLRA
FE44 20  03      3  1110          BRA END3
FE46 8B  95      2  1111  OKAY7   ADDA #$95         *SUBTRACT 5 TO OFFSET SENSOR FROM GROUND
FE48 19          2  1112          DAA               *THEN RESET BCD NUMBER
FE49 A7  05      5  1113  END3    STA 5,X           *STORE THE RESULT
FE4B 39          5  1114          RTS               *THEN RETURN TO CBCALC
```

```
                          1116  *  <<  SUBROUTINE CBCMPR >>
                          1117  *
                          1118  *       THIS SUBROUTINE UPDATES CB SETPOINT CHANGES ACCORDING
                          1119  *       TO ALLFGS AND INITIATES CB CONTROL AS NECESSARY
                          1120  *       THE CODE THAT IS NOT EXECUTED BECAUSE OF THE * IN FRONT
                          1121  *       IS THE PROGRAM PART THAT EXECUTES PULSE ACTION. PULSE
                          1122  *       ACTION CAN BE INITIATED BY REMOVING THE *'S. THIS PROGRAM
                          1123  *       IS WRITTEN WITH NO PULSE ACTION
                          1124  *
FE4C B6   D046      5     1125  CBCMPR   LDA ALLFGS
FE4F 85   40        2     1126           BITA #%01000000     *CHANGE CB SETPOINTS?
FE51 27   0A        3     1127           BEQ NCHANG          *NO GO TO CONTROL CB BLOCK
FE53 86   99        2     1128           LDA #$99            *YES CHANGE THE SETPOINT
FE55 B7   D042      5     1129           STA SPLIM           *LOAD LIMIT FLAG
FE58 C6   06        2     1130           LDB #$06            *SET UP B ACC. OFFSET
FE5A BD   FD74      8     1131           JSR SPCHNG          *JSR SETPOINT CHANGE
                          1132  *NCHANG JSR DOUBLE        *AVERAGE TWO READINGS
                          1133  *       TSTB              *CHECK IF CONTROL IS TO BE UPDATED
                          1134  *       BEQ ENDCB         *END IF NO CONTROL IS TO BE UPDATED
FE5D A6   05        5     1135  NCHANG   LDA 5,X             *GET *CBACT
FE5F A1   06        5     1136           CMPA 6,X            *ACT > SP?
FE61 2E   16        3     1137           BGT TOLER1          *YES GO CHECK THE TOLERANCE
FE63 A6   06        5     1138           LDA 6,X
FE65 A0   05        5     1139           SUBA 5,X            *(SP-ACT)>= 2?
FE67 81   01        2     1140           CMPA #$01
FE69 2F   22        3     1141           BLE ZERO            *NO DON't INITIATE CONTROL
                          1142  *        TST $C4,X           *THIS CODE IS USED WHEN
                          1143  *        BEQ G01             *PULSE ACTION IS INITIATED
                          1144  *        DEC $C4,X
                          1145  *        BRA ZERO
                          1146  *G01     INC $C4,X
FE6B A6   89 F000   8     1147           LDA -$1000,X        *SET RAISE BIT IN PIA
FE6F 84   38        2     1148           ANDA #%00111000
FE71 8A   40        2     1149           ORA #%01000000
FE73 A7   89 F000   8     1150           STA -$1000,X
FE77 20   1E        3     1151           BRA ENDCB           *THEN END
```

```
FE79 A0   06       5  1153  TOLER1   SUBA 6,X.          *(ACT-SP) > 2 ?
FE7B 81   01       2  1154           CMPA #$01
FE7D 2F   0E       3  1155           BLE ZERO           *NO DON'T INITIATE CONTROL
                      1156  *        TST $C4,X          *THIS CODE IS USED WHEN
                      1157  *        BEQ GO2            *PULSE ACTION IS INITIATED
                      1158  *        DEC $C4,X
                      1159  *        BRA ZERO
                      1160  *GO2     INC $C4,X
FE7F A6   89 F000  8  1161           LDA -$1000,X       *SET LOWER BIT IN PIA
FE83 84   B0       2  1162           ANDA #%10110000
FE85 8A   80       2  1163           ORA #%10000000
FE87 A7   89 F000  8  1164           STA -$1000,X
FE8B 20   0A       3  1165           BRA ENDCB          *THEN END
FE8D A6   89 F000  8  1166  ZERO     LDA -$1000,X       *SET RAISE AND LOWER BITS LOW IN PIA
FE91 84   30       2  1167           ANDA #%00110000
FE93 A7   89 F000  8  1168           STA -$1000,X
FE97 39            5  1169  ENDCB    RTS                *RETURN TO SUB AUTO
```

```
                          1171  *   <<  SUBROUTINE DOUBLE >>
                          1172  *
                          1173  *       THIS SUBROUTINE AVERAGES TWO CONSECUTIVE CB READINGS
                          1174  *       AND A CONTROL UPDATE FLAG IN THE B REGISTER
                          1175  *       WHICH IS USED IN AUTO MODE. THIS ROUTINE IS CALLED BY
                          1176  *       SUBROUTINE CBCMPR AND CBMAN WHEN PULSING ACTION IS
                          1177  *       REQUIRED
                          1178  *
FE98 5F             2     1179  DOUBLE  CLRB          *CLEAR THE CONTROL UPDATE FLAG
FE99 6D  89 00C1   10     1180          TST $C1,X
FE9D 26  12         3     1181          BNE ADD2
FE9F 6C  89 00C1   10     1182          INC $C1,X
FEA3 A6  05         5     1183          LDA 5,X
FEA5 A7  89 00C2    8     1184          STA $C2,X
FEA9 A6  89 00C3    8     1185          LDA $C3,X
FEAD A7  05         5     1186          STA 5,X
FEAF 20  15         3     1187          BRA ENDDB
FEB1 6A  89 00C1   10     1188  ADD2    DEC $C1,X
FEB5 5C             2     1189          INCB
FEB6 A6  05         5     1190          LDA 5,X
FEB8 AB  89 00C2    8     1191          ADDA $C2,X
FEBC 44             2     1192          LSRA
FEBD 8B  00         2     1193          ADDA #0
FEBF 19             2     1194          DAA
FEC0 A7  05         5     1195          STA 5,X
FEC2 A7  89 00C3    8     1196          STA $C3,X
FEC6 39             5     1197  ENDDB   RTS
```

```
                          1199  *  <<  SUBROUTINE MANUAL >>
                          1200  *
                          1201  *       THIS SUBROUTINE UPDATES MANUAL CONTROL OF THE
                          1202  *       REEL AND CUTTER BAR
                          1203  *
FEC7 86   01        2     1204  MANUAL   LDA #$01          *INITIALIZE MODPOS
FEC9 B7   D043      5     1205           STA MODPOS
FECC 8E   D000      3     1206           LDX #$D000        *LOAD SENSOR SET BASE ADDRESS
                          1207  *
                          1208  *      FIRST PROCESS THE REEL DATA
                          1209  *
FECF 86   FF        2     1210  MNNEXT   LDA #$FF          *STORE MANUAL REEL FLAG IN *RACT
FED1 A7   01        5     1211           STA 1,X
FED3 6F   84        6     1212           CLR ,X            *CLEAR THE REEL SUM FOR NEXT DATA
FED5 BD   FF18      8     1213  BRA12    JSR RMANUL        *JSR RMANUL AND INITIATE MANUAL CONTROL
                          1214  *
                          1215  *      SECONDLY, PROCESS CB DATA
                          1216  *
FED8 A6   07        5     1217  BRA11    LDA 7,X           *CHECK IF CB DATA IS VALID?
FEDA 4D             2     1218           TSTA
FEDB 27   08        3     1219           BEQ MCBERR        *IF ERROR GO TO MCBERR
FEDD BD   FD9D      8     1220           JSR CBCALC        *IF DATA IS VALID JSR CBCALC
FEE0 BD   FE98      8     1221           JSR DOUBLE        *AVERAGE THE LAST TWO VALUES IF IN PULSE MODE
FEE3 20   04        3     1222           BRA OVER
FEE5 86   FF        2     1223  MCBERR   LDA #$FF          *IF CB DATA IS INVALID LOAD ERROR FLAG
FEE7 A7   05        5     1224           STA 5,X           *INTO *CBACT
FEE9 E6   05        5     1225  OVER     LDB 5,X           *CHANGE THE ACTUAL CB DATA INTO
FEEB BD   FC00      8     1226           JSR SPLIT         *ASCII CHARACTERS SO THEY CAN BE
FEEE A7   88 52     5     1227           STA 82,X          *SENT TO TAPE STORAGE
FEF1 E7   88 53     5     1228           STB 83,X
FEF4 86   20        2     1229           LDA #BLANK        *INSERT BLANKS BETWEEN CHARACTERS
FEF6 A7   88 54     5     1230           STA 84,X
FEF9 6F   03        7     1231           CLR 3,X           *CLEAR CBSUM REGISTER FOR NEXT
FEFB 6F   04        7     1232           CLR 4,X           *DATA ACQUISITION PHASE
FEFD 6F   0A        7     1233           CLR 10,X
FEFF 6F   0B        7     1234           CLR 11,X
FF01 6F   0E        7     1235           CLR 14,X
FF03 6F   0F        7     1236           CLR 15,X
FF05 BD   FF4F      8     1237           JSR CBMAN         *INITIATE CB MANUAL CONTROL
FF08 86   08        2     1238           LDA #08
FF0A A7   07        5     1239           STA 7,X           *CLEAR CB ERROR FLAG
FF0C 30   88 10     5     1240           LEAX $10,X        *INC X FOR NEXT DATA SET
FF0F 78   D043      7     1241           LSL MODPOS        *INC SENSOR SET POINTER
FF12 8C   D040      4     1242           CMPX #$D040       *FINISHED ALL SENSOR SETS?
FF15 26   B8        3     1243           BNE MNNEXT        *NO
FF17 39             5     1244           RTS               *YES RETURN TO SUB COMPAR
```

```
                                1246  *   <<  SUBROUTINE RMANUL >>
                                1247  *
                                1248  *        THIS SUBROUTINE INITIATES MANUAL REEL CONTROL
                                1249  *        ACCORDING TO ALLFGS STATE: SUB IS CALLED FROM MANUAL
                                1250  *
FF18 B6   D046       5          1251  RMANUL   LDA ALLFGS
FF1B 85   20         2          1252           BITA #%00100000     *UPDATE REEL CONTROL?
FF1D 27   25         3          1253           BEQ IDOL            *IF NO THEN RESET RAISE AND LOWER BITS
FF1F B5   D043       5          1254           BITA MODPOS         *CHANGE THE REEL POSITION ?
FF22 27   20         3          1255           BEQ IDOL            *NO GO TO IDOL BLOCK
FF24 85   10         2          1256           BITA #%00010000     *RAISE ?
FF26 27   0E         3          1257           BEQ LMN             *NO LOWER GOTO LOWER BLOCK
FF28 A6   89 F000    8          1258           LDA -$1000,X        *SET RAISE BITS IN PIA
FF2C 84   D0         2          1259           ANDA #%11010000
FF2E 8A   10         2          1260           ORA #%00010000
FF30 A7   89 F000    8          1261           STA -$1000,X
FF34 20   1B         3          1262           BRA ENDRMA          *THEN END
FF36 A6   89 F000    8          1263  LMN      LDA -$1000,X        *SET LOWER BITS IN PIA
FF3A 84   E0         2          1264           ANDA #%11100000
FF3C 8A   20         2          1265           ORA #%00100000
FF3E A7   89 F000    8          1266           STA -$1000,X
FF42 20   0A         3          1267           BRA  ENDRMA         *THEN END
FF44 A6   89 F000    8          1268  IDOL     LDA -$1000,X        *CLEAR RAISE AND LOWER BITS
FF48 84   C0         2          1269           ANDA #%11000000     *THEN END
FF4A A7   89 F000    8          1270           STA -$1000,X
FF4E 39              5          1271  ENDRMA   RTS                 *RETURN TO SUB MANUAL
```

```
                        1273  *  <<  SUBROUTINE CBMAN >>
                        1274  *
                        1275  *       THIS SUBROUTINE INITIATES MANUAL CB CONTROL DEPENDING
                        1276  *       ON THE ALLFLG STATE. SUB CBMAN IS CALLED BY EITHER SUB
                        1277  *       MANUAL OR AUTO
                        1278  *
FF4F B6   D046      5   1279  CBMAN    LDA ALLFGS
FF52 85   40        2   1280           BITA #%01000000     *UPDATE CB MANUAL CONTROL ?
FF54 27   25        3   1281           BEQ CBIDOL          *NO THEN RESET RAISE AND LOWER BITS
FF56 B5   D043      5   1282           BITA MODPOS         *CHANGE CB HEIGHT ?
FF59 27   20        3   1283           BEQ CBIDOL          *NO GOTO CBIDOL BLOCK
FF5B 85   10        2   1284           BITA #%00010000     *RAIZE CB ?
FF5D 27   0E        3   1285           BEQ LCB             *NO GOTO LOWER CB BLOCK
FF5F A6   89 F000   8   1286           LDA -$1000,X        *SET RAISE BIT IN PIA
FF63 84   70        2   1287           ANDA #%01110000
FF65 8A   40        2   1288           ORA #%01000000
FF67 A7   89 F000   8   1289           STA -$1000,X
FF6B 20   18        3   1290           BRA ENDMOD          *THEN END
FF6D A6   89 F000   8   1291  LCB      LDA -$1000,X        *SET LOWER BIT IN PIA
FF71 84   B0        2   1292           ANDA #%10110000
FF73 8A   80        2   1293           ORA #%10000000
FF75 A7   89 F000   8   1294           STA -$1000,X
FF79 20   0A        3   1295           BRA ENDMOD          *THEN END
FF7B A6   89 F000   8   1296  CBIDOL   LDA -$1000,X        *CLEAR RAISE AND LOWER BITS IN PIA
FF7F 84   30        2   1297           ANDA #%00110000     *THEN END
FF81 A7   89 F000   8   1298           STA -$1000,X
FF85 39             5   1299  ENDMOD   RTS                 *THEN RETURN TO AUTO OR MANUAL
```

```
                        1301  *
                        1302  *  <<  SUBROUTINE SEND >>
                        1303  *
                        1304  *  THIS ROUTINE SENDS ASCII CHARACTERS THROUGH THE ACIA
                        1305  *  IN RS232 FORMAT.  THIS ROUTINE IS CALLED FROM SUB
                        1306  *  POLCR.
                        1307  *
                        1308  *
FF86 36   10        5   1309  SEND    PSHU X          *STORE THE X REGISTER ON THE STACK
FF88 8E   D080      3   1310          LDX #$D080      *LOAD THE INDEX REGISTER
FF8B F6   D072      5   1311          LDB OFFSET      *LOAD B ACC. WITH THE PROPER OFFSET
FF8E B6   E008      5   1312  BRA1    LDA ACIASR      *CHECK IF ACIA IS READY FOR NEW DATA
FF91 85   02        2   1313          BITA #%00000010
FF93 27   2C        3   1314          BEQ BRA3        *IF NOT READY THEN END
FF95 C1   08        2   1315          CMPB #08        *CHECK IF B >=8
FF97 2C   08        3   1316          BGE BRA4
FF99 A6   85        5   1317          LDA B,X         *ELSE SEND DATA TO ACIA
FF9B B7   E009      5   1318          STA ACIADR
FF9E 5C             2   1319          INCB
FF9F 20   1D        3   1320          BRA BRA2
FFA1 B6   D07A      5   1321  BRA4    LDA SCOUNT
FFA4 81   06        2   1322          CMPA #06        *IF SIX DATA SETS HAVE BEEN SENT,
FFA6 26   0A        3   1323          BNE BRA5        *THEN SEND A CARRAGE RETURN AND LINE FEED
FFA8 86   0D        2   1324          LDA #$0D        *SEND A CARRAGE RETURN
FFAA B7   E009      5   1325          STA ACIADR
FFAD 7C   D07A      7   1326          INC SCOUNT      *AND SET SCOUNT TO FLAG LINE FEED SEND
FFB0 20   0C        3   1327          BRA BRA2
FFB2 81   07        2   1328  BRA5    CMPA #07        *SEND A LINE FEED?
FFB4 26   08        3   1329          BNE BRA2
FFB6 7F   D07A      7   1330          CLR SCOUNT      *RESET SCOUNT FLAG
FFB9 86   0A        2   1331          LDA #$0A        *AND SEND A LINE FEED
FFBB B7   E009      5   1332          STA ACIADR
FFBE F7   D072      5   1333  BRA2    STB OFFSET      *BEFORE ENDING STORE X REGISTER,
FFC1 37   10        5   1334  BRA3    PULU X          *PREVIOUS X REGISTER, AND THE OFFSET
FFC3 39             5   1335          RTS             *THEN RETURN TO POLCR
0000                    1336          END
```

SYMBOL	VALUE	DEFN	REFERENCES								
ABCD	F9D7	449	202								
ACBACT	D005	103	558	562	591	595					
ACBERR	D007	101									
ACBSG2	D00E	97									
ACBSL2	D00F	96									
ACBSMH	D003	105									
ACBSML	D004	104									
ACBSP	D006	102	306	571	578						
ACHAR	0041	12	609	621	703	714					
ACIACR	E008	141	287	289							
ACIADR	E009	143	1318	1325	1332						
ACIASR	E008	142	1312								
ADD2	FEB1	1188	1181								
ADJUST	FA3F	505	203								
ALLFGS	D046	35	818	819	918	964	1125	1251	1279		
AMULTH	D008	100									
AMULTL	D00A	98									
AMULTM	D009	99									
ARACT	D001	107	599								
ARSP	D002	106	311	582	586						
ARSUM	D000	108									
ATPIAC	8003	123	223	240							
ATPIAD	8002	122	232	247	293	301	426	755	757	775	777
AUNEXT	FC84	838	876								
AUTMAN	F984	425	201								
AUTO	FC7C	832	822								
AUTO1	F9C5	433	429								
AUTO2	F9D1	439	436								
AUT1	FABB	567	549								
BACK	FC7B	825	823								
BACK1	FAA5	558	556								
BCBACT	D015	87									
BCBERR	D017	85									
BCBSH2	D01E	81									
BCBSL2	D01F	80									
BCBSMH	D013	89									
BCBSML	D014	88									
BCBSP	D016	86	307								
BCD	FDF1	1067	1038								
BCHAR	D042	13	632	716							
BEGIN	FB1E	195	207								
BLANK	0020	24	543	881	1229						
BMULTH	D01B	84									
BMULTL	D01A	82									
BMULTM	D019	83									
BRACT	D011	91									
BRA1	FFBE	1312									
BRA11	FED8	1217									
BRA12	FED5	1213									
BRA2	FFBE	1333	1320	1327	1329						
BRA3	FFC1	1334	1314								

SYMBOL	VALUE	DEFN	REFERENCES															
BRA4	FFA1	1321	1316															
BRA5	FFB2	1328	1323															
BRSP	D012	90	312															
BRSUM	D010	92																
CB	FC97	849	844															
CBCALC	FD90	991	852	1220														
CBCMPR	FE4C	1125	853															
CBDOWN	FA58	517	513															
CBERR	FCA4	855	851															
CBIDOL	FF7B	1286	1281	1283														
CBMAN	FF4F	1279	857	1237														
CCBACT	D025	70																
CCBERR	D027	68																
CCBSH2	D02E	64																
CCBSL2	D02F	63																
CCBSMH	D023	72																
CCBSML	D024	71																
CCBSP	D026	69	308															
CCHAR	0043	14	630	718														
CHCBSW	FA4E	512	509															
CHKCNT	FDC3	1016	1013															
CHRBUF	D045	32	704	710	724													
CHRLSW	FA64	523	511															
CMULTH	D028	67																
CMULTL	D02A	65																
CMULTM	D029	66																
COMPAR	FC59	811	306															
COUNT1	E002	135	368	408														
COUNT2	E004	136	362	369	409													
COUNT3	E006	137	370	397	410													
CRACT	D021	74																
CRSP	D022	73	313															
CRSUM	D020	75																
DCBACT	D035	54																
DCBERR	D037	52																
DCBSH2	D03E	48																
DCBSL2	D03F	47																
DCBSMH	D033	56																
DCBSML	D034	55																
DCBSP	D036	53	309															
DCHAR	0044	15	720															
DISBUF	D050	31	541	608	610	612	615	617	619	622	624	626	628	629	631	633	635	637
			639	649	650	654	655	659	660	664	665	675	679	683	687	715	717	719
			721	725	758													
DISPLY	FA7E	541	204															
DIVIDE	FD8A	1010	1017															
DIV16	FDE5	1048	992	995	999	1033												
DMULTH	D038	51																
DMULTL	D03A	49																
DMULTM	D039	50																
DONE	FA7A	534	516	521	527	532												

SYMBOL	VALUE	DEFN	REFERENCES						
DOUBLE	FE98	1179	1221						
DRACT	D031	58							
DRSP	D032	57	314						
DRSUM	D030	59							
ECHAR	0045	16	616						
ENDCAL	FCFC	902	900						
ENDCB	FE97	1169	1151	1165					
ENDDB	FEC6	1197	1187						
ENDMOD	FF85	1299	1290	1295					
ENDRCM	FD73	957	942	952					
ENDRMA	FF4E	1271	1262	1267					
ENDSP	FD8C	984	966	971	976	979			
END3	FE49	1113	1110						
ERROR	F93C	365	360						
ERROR2	F98A	404	395						
FINN2	FE3D	1106	1077						
FIRQ	F928	358	161						
FIXERR	FCFA	901	893	898					
FLAGS	F800	168	294						
FOUND	F8F4	724	708						
GET	FBC6	703	559	572	587	592	600		
GETCHR	0084	148	760						
IDOL	FF44	1268	1253	1255					
INACT	FAF4	590	569						
INCCSR	0089	147	768						
INIT	F842	217	191						
IRQ	F969	393	163						
LATCH1	E002	132	135	252	372	412			
LATCH2	E004	133	136	254	374	414			
LATCH3	E006	134	137	255	375	415			
LCB	FF6D	1291	1285						
LCD	0000	149	296	299	761	762	766	769	774
LCHAR	004C	17	638						
LDELAY	FC41	786							
LDFLG	F8C2	295	298						
LMN	FF36	1263	1257						
LOOP	FC46	788	789						
LOOPDS	FA85	544	547						
LOOPG	FBCF	707	713						
LOOP1	FC52	801	802						
LOWER	FD49	943	931						
LOWER1	FD55	947	944						
LSP	FD90	977	968						
MANUAL	FEC7	1204	824						
MAN3	FAA2	557	551	554					
MAN4	FABA	566							
MAN5	FAAF	562							
MCBERR	FEE5	1223	1219						
MCHAR	004D	18	607						
MLOOP	F823	197	200						
MN	FC78	824	821						

SYMBOL	VALUE	DEFN	REFERENCES															
MNNEXT	FECF	1210	1243															
MODFGS	D041	39	425	431	434	451	462	464	466	468	471	473	475	477	480	482	485	487
			490	492	494	496	505	534	548	552	567	575	705	814				
MODPOS	D043	37	833	874	965	1205	1241	1254	1282									
NCHANG	FE5D	1135	1127															
NCHAR	004E	19	611															
NOCB	FA62	522	518															
NOCBD	FADE	582	577															
NOCNTL	FD69	954	936	946														
NOCTL	FD67	953	930															
NOECHO	D047	34	338	342	376													
NOFF1	FAB7	565	561															
NOFF2	FAE6	585	574															
NOFF3	FB09	598	594															
NORL	FA78	533	529															
NOSPCH	FD22	928	920															
NOTSWA	F9F7	466	461															
NOTSWB	FA0C	475	470															
NOTSWC	FA21	485	479															
NOTSWD	FA36	494	489															
NXTCHR	FC1B	760	772															
NXTSET	F8FD	329	345															
OFFSET	D072	29	318	812	1311	1333												
OKAY2	FDF0	1058	1056															
OKAY3	FE15	1085	1081															
OKAY4	FE21	1091	1087															
OKAY5	FE2F	1099	1094															
OKAY6	FE38	1105	1101															
OKAY7	FE46	1111	1108															
OUT	FCAB	858	854															
OVER	FEE9	1225	1222															
OVER1	F98E	405	403															
OVERB	FC61	814																
PASS	FC6A	818	816															
PCOUNT	D040	40	196	197	199													
PIAACR	C001	113	218	234														
PIAADR	C000	112	113	225	242													
PIABCR	C011	115	219	235														
PIABDR	C010	114	115	226	243													
PIACCR	C021	117	220	236														
PIACDR	C020	116	117	227	244													
PIADCR	C031	119	221	237														
PIADDR	C030	118	119	228	245													
PIASWC	8001	121	222	239														
PIASWD	8000	120	121	122	123	230	449	506										
PIPELN	D044	36	316	1035														
POLCR	F8F8	327	198															
PTMCR1	E000	128	129	130	131	132	133	134	278									
PTMCR2	E001	129	269															
PTMCR3	E000	130	260															
PTMSR	E001	131	358	393														

SYMBOL	VALUE	DEFN	REFERENCES									
QUIT	F968	380	364									
RAISE	FD37	937	934									
RANCLR	F814	185	187									
RCALC	FCE8	892	838									
RCHAR	0052	20	636									
RCMPAR	FD11	918	843									
REELAM	D070	30	437	440	550	841						
REELCK	F9CA	435	432									
RERROR	FC94	845	840	842								
RESET	F80A	179	159									
RFRSH	FC0C	754	205									
RLDOWN	FA6E	528	524									
RMANUL	FF18	1251	845	1213								
RNDIRQ	F91D	342	339									
ROUND3	FDF9	1071	1105									
SAME	FC2C	768	764									
SCHAR	0053	21	614									
SCOUNT	D07A	28	813	1321	1326	1330						
SDELAY	FC4D	799	367	407	767							
SEND	FF86	1309	335									
SPCHNG	FD74	964	924	1131								
SPLIM	D042	38	922	970	1129							
SPLIT	FC00	739	648	653	658	663	674	678	682	686	727	859	1226
SSTACK	D600	153	180									
SWA	F9EA	460	453									
SWB	F9FF	469	455									
SWC	FA14	478	457									
SWD	FA29	488	459									
TCHAR	0054	22	618	625								
TEMP1	D048	33	363	378	379	400	416	417				
TOLER1	FE79	1153	1137									
UCHAR	0055	23	623									
USTACK	D7FF	154	179									
WRAUT	FB37	621	590									
WRCB	FB47	628	565	585	598							
WRCBAL	FB66	647	563	579	596							
WRMAN	FB17	607	557									
WRRL	FB58	635	588	601								
WRRLAL	FB9C	673	583									
WRSET	FB27	614	555	570								
ZERO	FE80	1166	1141	1155								

Motorola M6809 Assembler (UL202) done at 20:28:35 on 09-13-88

0 error(s) detected.

Cards:	1336	Symbols:	242	Cost:	$0.16
Punch:	0	References:	409	CPU Time:	0.48
Print:	1781	Storage:	32		

12. APPENDIX B. CUTTER BAR HEIGHT SENSOR TEST DATA

Table 12.1 Cutter bar height sensor test results over stubble terrain

			VERTICAL SENSOR				HORIZONTAL SENSOR			
Height (mm)	Speed (km/h)	Rep	Mean (mm)	CV (%)	Error Count	Sample Size	Mean (mm)	CV (%)	Error Count	Sample Size
130	5.0	1	96	29.7	4	87	96	24.7	0	91
		2	97	24.6	0	91	91	24.8	0	91
		3	90	36.0	2	89	89	30.3	1	90
	6.8	1	93	26.5	8	54	94	27.6	1	61
		2	84	41.4	3	59	90	26.2	1	61
		3	91	35.2	3	59	92	23.2	1	61
	12.0	1	97	28.2	1	41	91	27.4	0	42
		2	96	23.2	3	38	92	22.5	0	41
		3	90	29.0	2	39	94	20.7	1	40
380	5.0	1	351	4.2	0	81	341	4.7	0	81
		2	350	4.1	0	83	338	3.8	0	83
		3	350	4.2	0	82	340	3.6	0	82
	6.8	1	349	4.2	0	60	338	3.5	1	59
		2	351	4.2	0	60	338	3.7	1	59
		3	349	4.2	0	59	336	3.4	1	58
	12.0	1	353	4.9	0	33	338	3.5	1	32
		2	347	5.9	0	33	335	3.5	0	33
		3	352	5.2	0	32	338	3.4	1	31
740	5.0	1	696	2.3	3	71	704	2.0	0	74
		2	691	2.4	4	69	701	2.2	1	72
		3	691	2.5	2	70	700	2.0	1	71
	6.8	1	696	2.2	4	47	703	2.2	1	50
		2	693	2.7	2	52	703	2.0	1	53
		3	693	2.6	1	53	703	2.2	0	54
	12.0	1	691	3.0	1	31	700	1.8	1	31
		2	694	1.7	2	29	702	1.7	1	30
		3	689	3.0	0	32	701	2.0	1	31

Table 12.2 Cutter bar height sensor test results over hay land terrain

			VERTICAL SENSOR				HORIZONTAL SENSOR			
Height (mm)	Speed (km/h)	Rep	Mean (mm)	CV (%)	Error Count	Sample Size	Mean (mm)	CV (%)	Error Count	Sample Size
		1	39	98.2	0	83	22	130.0	0	83
	5.0	2	38	97.7	0	82	21	127.1	1	81
		3	40	98.0	0	81	22	123.5	1	80
		1	31	122.9	0	60	18	118.7	1	59
130	6.8	2	44	100.9	0	60	18	113.2	1	59
		3	37	99.4	0	59	15	158.5	1	58
		1	30	106.1	0	34	18	163.7	1	33
	12.0	2	34	119.2	0	33	16	118.1	1	32
		3	35	96.1	0	34	10	164.6	1	33
		1	331	15.4	6	75	321	8.2	1	80
	5.0	2	341	5.0	4	76	315	10.7	1	79
		3	339	8.2	2	77	318	8.7	1	78
		1	335	15.9	1	56	316	9.3	1	56
380	6.8	2	334	12.4	1	56	321	8.6	1	56
		3	340	5.9	1	56	320	10.0	1	56
		1	335	3.6	1	31	318	11.6	1	31
	12.0	2	340	6.6	2	30	317	8.8	1	31
		3	342	4.6	3	29	314	10.0	1	31
		1	684	4.7	11	65	676	6.0	0	76
	5.0	2	673	7.3	8	67	674	6.4	1	74
		3	679	7.9	11	64	674	5.9	1	74
		1	683	5.1	0	40	672	5.7	1	39
740	6.8	2	673	6.2	4	50	669	5.6	1	53
		3	677	5.5	2	52	665	5.7	1	53
		1	675	5.0	1	30	668	4.6	1	30
	12.0	2	679	3.8	3	28	662	5.6	1	30
		3	673	8.0	4	27	675	3.7	1	30